

example, the format of the EVENT method map MDE to be read and accessed immediately subsequently (by blocks 1950, 1952). In the preferred embodiment, MDEs and UDEs may have any of various different formats, and their formats may be flexibly  
5 specified or changed dynamically depending upon the installation, user, etc. The DTD, in effect, describes to the EVENT method 1940 how to read from the EVENT method map MDE. DTDs are also used to specify how methods should write to MDEs and UDEs, and thus may be used to implement privacy  
10 filters by, for example, preventing certain confidential user information from being written to data structures that will be reported to third parties.

Block 1950 ("map event to atomic element # and event  
15 count using a Map MDE") is in some sense the "heart" of EVENT method 1940. This step "maps" the event into an "atomic element number" to be responded to by subsequently called methods. An example of process control steps performed by a somewhat representative example of this "mapping" step 1950 is  
20 shown in Figure 53b.

The Figure 53b example shows the process of converting a READ event that is associated with requesting byte range 1001-1500 from a specific piece of content into an appropriate atomic

element. The example EVENT method mapping process (block 1950 in Figure 53a) can be detailed as the representative process shown in Figure 53b.

5           EVENT method mapping process 1950 may first look up  
the event code (READ) in the EVENT method MDE (1952) using  
the EVENT method map DTD (1948) to determine the structure  
and contents of the MDE. A test might then be performed to  
determine if the event code was found in the MDE (1956), and if  
10 not ("No" branch), the EVENT method mapping process may the  
terminate (1958) without mapping the event to an atomic  
element number and count. If the event was found in the MDE  
("Yes" branch), the EVENT method mapping process may then  
compare the event range (e.g., bytes 1001-1500) against the  
15 atomic element to event range mapping table stored in the MDE  
(block 1960). The comparison might yield one or more atomic  
element numbers or the event range might not be found in the  
mapping table. The result of the comparison might then be  
tested (block 1962) to determine if any atomic element numbers  
20 were found in the table. If not ("No" branch), the EVENT  
method mapping process may terminate without selecting any  
atomic element numbers or counts (1964). If the atomic element  
numbers were found, the process might then calculate the atomic  
element count from the event range (1966). In this example, the

process might calculate the number of bytes requested by subtracting the upper byte range from the lower byte range (e.g.,  $1500 - 1001 + 1 = 500$ ). The example EVENT method mapping process might then terminate (block 1968) and return the atomic  
5 element number(s) and counts.

EVENT method 1940 may then write an EVENT audit trail if required to an EVENT method Audit Trail UDE (block 1970, 1972). EVENT method 1940 may then prepare to pass the  
10 atomic element number and event count to the calling CONTROL method (or other control process) (at exit point 1978). Before that, however, EVENT method 1940 may test whether an atomic element was selected (decision block 1974). If no atomic element was selected, then the EVENT method may be failed  
15 (block 1974). This may occur for a number of reasons. For example, the EVENT method may fail to map an event into an atomic element if the user is not authorized to access the specific areas of content that the EVENT method MDE does not describe. This mechanism could be used, for example, to distribute  
20 customized versions of a piece of content and control access to the various versions in the content object by altering the EVENT method MDE delivered to the user. A specific use of this technology might be to control the distribution of different

language (e.g., English, French, Spanish) versions of a piece of content.

### **Billing**

5           Figure 53c is a flowchart of an example of process control steps performed by a BILLING method 1980. Examples of BILLING methods are set forth in Figures 49d, 50d, and 51d and are described above. BILLING method 1980 shown in Figure 53c is somewhat more generalized than the examples above. Like  
10           the BILLING method examples above, BILLING method 1980 receives a meter value to determine the amount to bill.

BILLING method 1980 may first prime a BILLING audit trail (if required) by writing appropriate information to the BILLING method Audit Trail UDE (blocks 1982, 1984). BILLING method  
15           1980 may then obtain and load a BILLING method map DTD from the secure database (blocks 1985, 1986), which describes the BILLING method map MDE (e.g., a price list, table, or parameters to the billing amount calculation algorithm) that should be used by this BILLING method. The BILLING method  
20           map MDE may be delivered either as part of the content object or as a separately deliverable component that is combined with the control information at registration.



The BILLING method map MDE in this example may describe the pricing algorithm that should be used in this BILLING method (e.g., bill \$0.001 per byte of content released). Block 1988 ("Map meter value to billing amount") functions in the same manner as block 1950 of the EVENT method; it maps the meter value to a billing value. Process step 1988 may also interrogate the secure database (as limited by the privacy filter) to determine if other objects or information (e.g., user information) are present as part of the BILLING method algorithm.

BILLING method 1980 may then write a BILLING audit trail if required to a BILLING method Audit Trail UDE (block 1990, 1992), and may prepare to return the billing amount to the calling CONTROL method (or other control process). Before that, however, BILLING method 1980 may test whether a billing amount was determined (decision block 1994). If no billing amount was determined, then the BILLING method may be failed (block 1996). This may occur if the user is not authorized to access the specific areas of the pricing table that the BILLING method MDE describes (e.g., you may purchase not more than \$100.00 of information from this content object).

### Access

Figure 54 is a flowchart of an example of program control steps performed by an ACCESS method 2000. As described above, an ACCESS method may be used to access content  
5 embedded in an object 300 so it can be written to, read from, or otherwise manipulated or processed. In many cases, the ACCESS method may be relatively trivial since the object may, for example, be stored in a local storage that is easily accessible. However, in the general case, an ACCESS method 2000 must go  
10 through a more complicated procedure in order to obtain the object. For example, some objects (or parts of objects) may only be available at remote sites or may be provided in the form of a real-time download or feed (e.g., in the case of broadcast transmissions). Even if the object is stored locally to the VDE  
15 node, it may be stored as a secure or protected object so that it is not directly accessible to a calling process. ACCESS method 2000 establishes the connections, routings, and security requisites needed to access the object. These steps may be performed transparently to the calling process so that the calling  
20 process only needs to issue an access request and the particular ACCESS method corresponding to the object or class of objects handles all of the details and logistics involved in actually accessing the object.

ACCESS method 2000 may first prime an ACCESS audit trail (if required) by writing to an ACCESS Audit Trail UDE (blocks 2002, 2004). ACCESS method 2000 may then read and load an ACCESS method DTD in order to determine the format of an ACCESS MDE (blocks 2006, 2008). The ACCESS method MDE specifies the source and routing information for the particular object to be accessed in the preferred embodiment. Using the ACCESS method DTD, ACCESS method 2000 may load the correction parameters (e.g., by telephone number, account ID, password and/or a request script in the remote resource dependent language).

ACCESS method 2000 reads the ACCESS method MDE from the secure database, reads it in accordance with the ACCESS method DTD, and loads encrypted content source and routing information based on the MDE (blocks 2010, 2012). This source and routing information specifies the location of the encrypted content. ACCESS method 2000 then determines whether a connection to the content is available (decision block 2014). This "connection" could be, for example, an on-line connection to a remote site, a real-time information feed, or a path to a secure/protected resource, for example. If the connection to the content is not currently available ("No" exit of decision block 2014), then ACCESS method 2000 takes steps to

open the connection (block 2016). If the connection fails (e.g., because the user is not authorized to access a protected secure resource), then the ACCESS method 2000 returns with a failure indication (termination point 2018). If the open connection  
5 succeeds, on the other hand, then ACCESS method 2000 obtains the encrypted content (block 2020). ACCESS method 2000 then writes an ACCESS audit trail if required to the secure database ACCESS method Audit Trail UDE (blocks 2022, 2024), and then terminates (terminate point 2026).

10

### **Decrypt and Encrypt**

Figure 55a is a flowchart of an example of process control steps performed by a representative example of a DECRYPT method 2030 provided by the preferred embodiment. DECRYPT  
15 method 2030 in the preferred embodiment obtains or derives a decryption key from an appropriate PERC 808, and uses it to decrypt a block of encrypted content. DECRYPT method 2030 is passed a block of encrypted content or a pointer to where the encrypted block is stored. DECRYPT 2030 selects a key number  
20 from a key block (block 2032). For security purposes, a content object may be encrypted with more than one key. For example, a movie may have the first 10 minutes encrypted using a first key, the second 10 minutes encrypted with a second key, and so on. These keys are stored in a PERC 808 in a structure called a "key

block." The selection process involves determining the correct key to use from the key block in order to decrypt the content. The process for this selection is similar to the process used by EVENT methods to map events into atomic element numbers.

5     DECRYPT method 2030 may then access an appropriate PERC 808 from the secure database 610 and loads a key (or "seed") from a PERC (blocks 2034, 2036). This key information may be the actual decryption key to be used to decrypt the content, or it may be information from which the decryption key may be at

10    least in part derived or calculated. If necessary, DECRYPT method 2030 computes the decryption key based on the information read from PERC 808 at block 2034 (block 2038). DECRYPT method 2030 then uses the obtained and/or calculated decryption key to actually decrypt the block of encrypted

15    information (block 2040). DECRYPT method 2030 outputs the decrypted block (or the pointer indicating where it may be found), and terminates (termination point 2042).

Figure 55b is a flowchart of an example of process control

20    steps performed by a representative example of an ENCRYPT method 2050. ENCRYPT method 2050 is passed as an input, a block of information to encrypt (or a pointer indicating where it may be found). ENCRYPT method 2050 then may determine an encryption key to use from a key block (block 2052). The

encryption key selection makes a determination if a key for a specific block of content to be written already exists in a key block stored in PERC 808. If the key already exists in the key block, then the appropriate key number is selected. If no such key exists in the key block, a new key is calculated using an algorithm appropriate to the encryption algorithm. This key is then stored in the key block of PERC 808 so that DECRYPT method 2030 may access the key in order to decrypt the content stored in the content object. ENCRYPT method 2050 then accesses the appropriate PERC to obtain, derive and/or compute an encryption key to be used to encrypt the information block (blocks 2054, 2056, 2058, which are similar to Figure 55a blocks 2034, 2036, 2038). ENCRYPT method 2050 then actually encrypts the information block using the obtained and/or derived encryption key (block 2060) and outputs the encrypted information block or a pointer where it can be found before terminating (termination point 2062).

### **Content**

Figure 56 is a flowchart of an example of process control steps performed by a representative of a CONTENT method 2070 provided by the preferred embodiment. CONTENT method 2070 in the preferred embodiment builds a "synopsis" of protected content using a secure process. For example,

CONTENT method 2070 may be used to derive unsecure  
("public") information from secure content. Such derived public  
information might include, for example, an abstract, an index, a  
table of contents, a directory of files, a schedule when content  
5 may be available, or excerpts such as for example, a movie  
"trailer."

CONTENT method 2070 begins by determining whether  
the derived content to be provided must be derived from secure  
10 contents, or whether it is already available in the object in the  
form of static values (decision block 2070). Some objects may, for  
example, contain prestored abstracts, indexes, tables of contents,  
etc., provided expressly for the purpose of being extracted by the  
CONTENT method 2070. If the object contains such static  
15 values ("static" exit to decision block 2072), then CONTENT  
method 2070 may simply read this static value content  
information from the object (block 2074), optionally decrypt, and  
release this content description (block 2076). If, on the other  
hand, CONTENT method 2070 must derive the synopsis/content  
20 description from the secure object ("derived" exit to decision block  
2072), then the CONTENT method may then securely read  
information from the container according to a synopsis algorithm  
to produce the synopsis (block 2078).

## Extract and Embed

Figure 57a is a flowchart of an example of process control steps performed by a representative example of an EXTRACT method 2080 provided by the preferred embodiment. EXTRACT method 2080 is used to copy or remove content from an object and place it into a new object. In the preferred embodiment, the EXTRACT method 2080 does not involve any release of content, but rather simply takes content from one container and places it into another container, both of which may be secure. Extraction of content differs from release in that the content is never exposed outside a secure container. Extraction and Embedding are complementary functions; extract takes content from a container and creates a new container containing the extracted content and any specified control information associated with that content. Embedding takes content that is already in a container and stores it (or the complete object) in another container directly and/or by reference, integrating the control information associated with existing content with those of the new content.

EXTRACT method 2080 begins by priming an Audit UDE (blocks 2082, 2084). EXTRACT method then calls a BUDGET method to make sure that the user has enough budget for (and is authorized to) extract content from the original object (block



2086). If the user's budget does not permit the extraction ("no" exit to decision block 2088), then EXTRACT method 2080 may write a failure audit record (block 2090), and terminate (termination point 2092). If the user's budget permits the extraction ("yes" exit to decision block 2088), then the EXTRACT method 2080 creates a copy of the extracted object with specified rules and control information (block 2094). In the preferred embodiment, this step involves calling a method that actually controls the copy. This step may or may not involve decryption and encryption, depending on the particular the PERC 808 associated with the original object, for example. EXTRACT method 2080 then checks whether any control changes are permitted by the rights authorizing the extract to begin with (decision block 2096). In some cases, the extract rights require an exact copy of the PERC 808 associated with the original object (or a PERC included for this purpose) to be placed in the new (destination) container ("no" exit to decision block 2096). If no control changes are permitted, then extract method 2080 may simply write audit information to the Audit UDE (blocks 2098, 2100) before terminating (terminate point 2102). If, on the other hand, the extract rights permit the user to make control changes ("yes" to decision block 2096), then EXTRACT method 2080 may call a method or load module that solicits new or changed control information (e.g., from the user, the distributor who

created/granted extract rights, or from some other source) from the user (blocks 2104, 2106). EXTRACT method 2080 may then call a method or load module to create a new PERC that reflects these user-specified control information (block 2104). This new  
5 PERC is then placed in the new (destination) object, the auditing steps are performed, and the process terminates.

Figure 57b is an example of process control steps performed by a representative example of an EMBED method  
10 2110 provided by the preferred embodiment. EMBED method 2110 is similar to EXTRACT method 2080 shown in Figure 57a. However, the EMBED method 2110 performs a slightly different function—it writes an object (or reference) into a destination container. Blocks 2112-2122 shown in Figure 57b are similar to  
15 blocks 2082-2092 shown in Figure 57a. At block 2124, EMBED method 2110 writes the source object into the destination container, and may at the same time extract or change the control information of the destination container. One alternative is to simply leave the control information of the destination  
20 container alone, and include the full set of control information associated with the object being embedded in addition to the original container control information. As an optimization, however, the preferred embodiment provides a technique whereby the control information associated with the object being

embedded are "abstracted" and incorporated into the control information of the destination container. Block 2124 may call a method to abstract or change this control information. EMBED method 2110 then performs steps 2126-2130 which are similar to  
5 steps 2096, 2104, 2106 shown in Figure 57a to allow the user, if authorized, to change and/or specify control information associated with the embedded object and/or destination container. EMBED method 2110 then writes audit information into an Audit UDE (blocks 2132, 2134), before terminating (at  
10 termination point 2136).

### Obscure

Figure 58a is a flowchart of an example of process control steps performed by a representative example of an OBSCURE  
15 method 2140 provided by the preferred embodiment. OBSCURE method 2140 is typically used to release secure content in devalued form. For example, OBSCURE method 2140 may release a high resolution image in a lower resolution so that a viewer can appreciate the image but not enjoy its full value. As  
20 another example, the OBSCURE method 2140 may place an obscuring legend (e.g., "COPY," "PROOF," etc.) across an image to devalue it. OBSCURE method 2140 may "obscure" text, images, audio information, or any other type of content.

OBSCURE method 2140 first calls an EVENT method to determine if the content is appropriate and in the range to be obscured (block 2142). If the content is not appropriate for obscuring, the OBSCURE method terminates (decision block 2144 "no" exit, terminate point 2146). Assuming that the content is to be obscured ("yes" exit to decision block 2144), then OBSCURE method 2140 determines whether it has previously been called to obscure this content (decision block 2148). Assuming the OBSCURE 2140 has not previously called for this object/content ("yes" exit to decision block 2148), the OBSCURE method 2140 reads an appropriate OBSCURE method MDE from the secure database and loads an obscure formula and/or pattern from the MDE (blocks 2150, 2152). The OBSCURE method 2140 may then apply the appropriate obscure transform based on the patterns and/or formulas loaded by block 2150 (block 2154). The OBSCURE method then may terminate (terminate block 2156).

### **Fingerprint**

Figure 58b is a flowchart of an example of process control steps performed by a representative example of a FINGERPRINT method 2160 provided by the preferred embodiment. FINGERPRINT method 2160 in the preferred embodiment operates to "mark" released content with a "fingerprint" identification of who released the content and/or

check for such marks. This allows one to later determine who released unsecured content by examining the content.

5 FINGERPRINT method 2160 may, for example, insert a user ID within a datastream representing audio, video, or binary format information. FINGERPRINT method 2160 is quite similar to OBSCURE method 2140 shown in Figure 58a except that the transform applied by FINGERPRINT method block 2174 "fingerprints" the released content rather than obscuring it.

10 Figure 58c shows an example of a "fingerprinting" procedure 2160 that inserts into released content "fingerprints" 2161 that identify the object and/or property and/or the user that requested the released content and/or the date and time of the release and/or other identification criteria of the released  
15 content.

Such fingerprints 2161 can be "buried" -- that is inserted in a manner that hides the fingerprints from typical users, sophisticated "hackers," and/or from all users, depending on the  
20 file format, the sophistication and/or variety of the insertion algorithms, and on the availability of original, non-fingerprinted content (for comparison for reverse engineering of algorithm(s)). Inserted or embedded fingerprints 2161, in a preferred embodiment, may be at least in part encrypted to make them

more secure. Such encrypted fingerprints 2161 may be embedded within released content provided in "clear" (plaintext) form.

5           Fingerprints 2161 can be used for a variety of purposes including, for example, the often related purposes of proving misuse of released materials and proving the source of released content. Software piracy is a particularly good example where fingerprinting can be very useful. Fingerprinting can also help  
10 to enforce content providers' rights for most types of electronically delivered information including movies, audio recordings, multimedia, information databases, and traditional "literary" materials. Fingerprinting is a desirable alternative or  
15 addition to copy protection.

Most piracy of software applications, for example, occurs not with the making of an illicit copy by an individual for use on another of the individual's own computers, but rather in giving a copy to another party. This often starts a chain (or more  
20 accurately a pyramid) of illegal copies, as copies are handed from individual to individual. The fear of identification resulting from the embedding of a fingerprint 2161 will likely dissuade most individuals from participating, as many currently do, in widespread, "casual" piracy. In some cases, content may be

checked for the presence of a fingerprint by a fingerprint method to help enforce content providers' rights.

5 Different fingerprints 2161 can have different levels of security (e.g., one fingerprint 2161(1) could be readable/identifiable by commercial concerns, while another fingerprint 2161(2) could be readable only by a more trusted agency. The methods for generating the more secure fingerprint 2161 might employ more complex encryption techniques (e.g.,  
10 digital signatures) and/or obscuring of location methodologies. Two or more fingerprints 2161 can be embedded in different locations and/or using different techniques to help protect fingerprinted information against hackers. The more secure fingerprints might only be employed periodically rather than  
15 each time content release occurs, if the technique used to provide a more secure fingerprint involves an undesired amount of additional overhead. This may nevertheless be effective since a principal objective of fingerprinting is deterrence—that is the fear on the part of the creator of an illicit copy that the copying  
20 will be found out.

For example, one might embed a copy of a fingerprint 2161 which might be readily identified by an authorized party—for example a distributor, service personal, client administrator, or

clearinghouse using a VDE electronic appliance 600. One might embed one or more additional copies or variants of a fingerprint 2161 (e.g., fingerprints carrying information describing some or all relevant identifying information) and this additional one or  
5 more fingerprints 2161 might be maintained in a more secure manner.

Fingerprinting can also protect privacy concerns. For example, the algorithm and/or mechanisms needed to identify  
10 the fingerprint 2161 might be available only through a particularly trusted agent.

Fingerprinting 2161 can take many forms. For example, in an image, the color of every N pixels (spread across an image, or spread across a subset of the image) might be subtly shifted in  
15 a visually unnoticeable manner (at least according to the normal, unaided observer). These shifts could be interpreted by analysis of the image (with or without access to the original image), with each occurrence or lack of occurrence of a shift in color (or  
20 greyscale) being one or more binary "on or off" bits for digital information storage. The N pixels might be either consistent, or alternatively, pseudo-random in order (but interpretable, at least in part, by a object creator, object provider, client administrator, and/or VDE administrator).



Other modifications of an image (or moving image, audio, etc.) which provide a similar benefit (that is, storing information in a form that is not normally noticeable as a result of a certain modification of the source information) may be appropriate, depending on the application. For example, certain subtle modifications in the frequency of stored audio information can be modified so as to be normally unnoticeable to the listener while still being readable with the proper tools. Certain properties of the storage of information might be modified to provide such slight but interpretable variations in polarity of certain information which is optically stored to achieve similar results. Other variations employing other electronic, magnetic, and/or optical characteristic may be employed.

Content stored in files that employ graphical formats, such as Microsoft Windows word processing files, provide significant opportunities for "burying" a fingerprint 2161. Content that includes images and/or audio provides the opportunity to embed fingerprints 2161 that may be difficult for unauthorized individuals to identify since, in the absence of an "unfingerprinted" original for purposes of comparison, minor subtle variations at one or more time instances in audio frequencies, or in one or more video images, or the like, will be in themselves undiscernible given the normally unknown nature of

the original and the large amounts of data employed in both image and sound data (and which is not particularly sensitive to minor variations). With formatted text documents, particularly those created with graphical word processors (such as Microsoft Windows or Apple MacIntosh word processors and their DOS and Unix equivalents), fingerprints 2161 can normally be inserted unobtrusively into portions of the document data representation that are not normally visible to the end user (such as in a header or other non-displayed data field).

10

Yet another form of fingerprinting, which may be particularly suitable for certain textual documents, would employ and control the formation of characters for a given font. Individual characters may have a slightly different visual formation which connotes certain "fingerprint" information. This alteration of a given character's form would be generally undiscernible, in part because so many slight variations exist in versions of the same font available from different suppliers, and in part because of the smallness of the variation. For example, in a preferred embodiment, a program such as Adobe Type Align could be used which, in its off-the-shelf versions, supports the ability of a user to modify font characters in a variety of ways. The mathematical definition of the font character is modified according to the user's instructions to produce a specific set of

15

20

modifications to be applied to a character or font. Information content could be used in an analogous manner (as an alternative to user selections) to modify certain or all characters too subtly for user recognition under normal circumstances but which nevertheless provide appropriate encoding for the fingerprint 2161. Various subtly different versions of a given character might be used within a single document so as to increase the ability to carry transaction related font fingerprinted information.

Some other examples of applications for fingerprinting might include:

1. In software programs, selecting certain interchangeable code fragments in such a way as to produce more or less identical operation, but on analysis, differences that detail fingerprint information.
2. With databases, selecting to format certain fields, such as dates, to appear in different ways.
3. In games, adjusting backgrounds, or changing order of certain events, including noticeable or very subtle changes in timing and/or ordering of appearance of game elements, or slight changes in the look of elements of the game.

Fingerprinting method 2160 is typically performed (if at all) at the point at which content is released from a content object 300. However, it could also be performed upon distribution of an object to "mark" the content while still in encrypted form. For example, a network-based object repository could embed fingerprints 2161 into the content of an object before transmitting the object to the requester, the fingerprint information could identify a content requester/end user. This could help detect "spoof" electronic appliances 600 used to release content without authorization.

### **Destroy**

Figure 59 is a flowchart of an example of process control steps performed by a representative performed by a DESTROY method 2180 provided by the preferred embodiment. DESTROY method 2180 removes the ability of a user to use an object by destroying the URT the user requires to access the object. In the preferred embodiment, a DESTROY method 2180 may first write audit information to an Audit UDE (blocks 2182, 2184). DESTROY method 2180 may then call a WRITE and/or ACCESS method to write information which will corrupt (and thus destroy) the header and/or other important parts of the object (block 2186). DESTROY method 2180 may then mark one or more of the control structures (e.g., the URT) as damaged by

writing appropriate information to the control structure (blocks 2188, 2190). DESTROY method 2180, finally, may write additional audit information to Audit UDE (blocks 2192, 2194) before terminating (terminate point 2196).

5

### **Panic**

Figure 60 is a flowchart of an example of process control steps performed by a representative example of a PANIC method 2200 provided by the preferred embodiment. PANIC method 2200 may be called when a security violation is detected. PANIC method 2200 may prevent the user from further accessing the object currently being accessed by, for example, destroying the channel being used to access the object and marking one or more of the control structures (e.g., the URT) associated with the user and object as damaged (blocks 2206, and 2208-2210, respectively). Because the control structure is damaged, the VDE node will need to contact an administrator to obtain a valid control structure(s) before the user may access the same object again. When the VDE node contacts the administrator, the administrator may request information sufficient to satisfy itself that no security violation occurred, or if a security violation did occur, take appropriate steps to ensure that the security violation is not repeated.

10

15

20

**Meter**

Figure 61 is a flowchart of an example of process control steps performed by a representative example of a METER method provided by the preferred embodiment. Although  
5 METER methods were described above in connection with Figures 49, 50 and 51, the METER method 2220 shown in Figure 61 is possibly a somewhat more representative example. In the preferred embodiment, METER method 2220 first primes an Audit Trail by accessing a METER Audit Trail UDE (blocks  
10 2222, 2224). METER method 2220 may then read the DTD for the Meter UDE from the secure database (blocks 2226, 2228). METER method 2220 may then read the Meter UDE from the secure database (blocks 2230, 2232). METER method 2220 next may test the obtained Meter UDE to determine whether it has  
15 expired (decision block 2234). In the preferred embodiment, each Meter UDE may be marked with an expiration date. If the current date/time is later than the expiration date of the Meter UDE ("yes" exit to decision block 2234), then the METER method 2220 may record a failure in the Audit Record and terminate  
20 with a failure condition (block 2236, 2238).

Assuming the Meter UDE is not yet expired, the meter method 2220 may update it using the atomic element and event count passed to the METER method from, for example, an

EVENT method (blocks 2239, 2240). The METER method 2220 may then save the Meter Use Audit Record in the Meter Audit Trail UDE (blocks 2242, 2244), before terminating (at terminate point 2246).

5

#### **Additional Security Features Provided by the Preferred Embodiment**

VDE 100 provided by the preferred embodiment has  
10 sufficient security to help ensure that it cannot be compromised short of a successful "brute force attack," and so that the time and cost to succeed in such a "brute force attack" substantially exceeds any value to be derived. In addition, the security provided by VDE 100 compartmentalizes the internal workings  
15 of VDE so that a successful "brute force attack" would compromise only a strictly bounded subset of protected information, not the entire system.

The following are among security aspects and features  
20 provided by the preferred embodiment:

- security of PPE 650 and the processes it performs
- security of secure database 610
- security of encryption/decryption performed by PPE 650

- key management; security of encryption/decryption keys and shared secrets
- security of authentication/external communications
- security of secure database backup
- 5      • secure transportability of VDE internal information between electronic appliances 600
- security of permissions to access VDE secure information
- security of VDE objects 300
- 10     • integrity of VDE security.

Some of these security aspects and considerations are discussed above. The following provides an expanded discussion of preferred embodiment security features not fully addressed elsewhere.

### **Management of Keys and Shared Secrets**

VDE 100 uses keys and shared secrets to provide security. The following key usage features are provided by the preferred embodiment:

- different cryptosystem/key types
- secure key length
- key generation
- key "convolution" and key "aging."



Each of these types are discussed below.

#### A. Public-Key and Symmetric Key Cryptosystems

5       The process of disguising or transforming information to  
hide its substance is called encryption. Encryption produces  
"ciphertext." Reversing the encryption process to recover the  
substance from the ciphertext is called "decryption." A  
cryptographic algorithm is the mathematical function used for  
encryption and decryption.

10

Most modern cryptographic algorithms use a "key." The  
"key" specifies one of a family of transformations to be provided.  
Keys allow a standard, published and tested cryptographic  
algorithm to be used while ensuring that specific  
15       transformations performed using the algorithm are kept secret.  
The secrecy of the particular transformations thus depends on  
the secrecy of the key, not on the secrecy of the algorithm.

20       There are two general forms of key-based algorithms,  
either or both of which may be used by the preferred embodiment  
PPE 650:

symmetric; and  
public-key ("PK").

Symmetric algorithms are algorithms where the encryption key can be calculated from the decryption key and vice versa. In many such systems, the encryption and decryption keys are the same. The algorithms, also called "secret-key",  
5 "single key" or "shared secret" algorithms, require a sender and receiver to agree on a key before ciphertext produced by a sender can be decrypted by a receiver. This key must be kept secret. The security of a symmetric algorithm rests in the key: divulging the key means that anybody could encrypt and decrypt  
10 information in such a cryptosystem. See Schneier, Applied Cryptography at Page 3. Some examples of symmetric key algorithms that the preferred embodiment may use include DES, Skipjack/Clipper, IDEA, RC2, and RC4.

15 In public-key cryptosystems, the key used for encryption is different from the key used for decryption. Furthermore, it is computationally infeasible to derive one key from the other. The algorithms used in these cryptosystems are called "public key" because one of the two keys can be made public without  
20 endangering the security of the other key. They are also sometimes called "asymmetric" cryptosystems because they use different keys for encryption and decryption. Examples of public-key algorithms include RSA, El Gamal and LUC.

The preferred embodiment PPE 650 may operate based on only symmetric key cryptosystems, based on public-key cryptosystems, or based on both symmetric key cryptosystems and public-key cryptosystems. VDE 100 does not require any specific encryption algorithms; the architecture provided by the preferred embodiment may support numerous algorithms including PK and/or secret key (non PK) algorithms. In some cases, the choice of encryption/decryption algorithm will be dependent on a number of business decisions such as cost, market demands, compatibility with other commercially available systems, export laws, etc.

Although the preferred embodiment is not dependent on any particular type of cryptosystem or encryption/decryption algorithm(s), the preferred example uses PK cryptosystems for secure communications between PPEs 650, and uses secret key cryptosystems for "bulk" encryption/decryption of VDE objects 300. Using secret key cryptosystems (e.g., DES implementations using multiple keys and multiple passes, Skipjack, RC2, or RC4) for "bulk" encryption/decryption provides efficiencies in encrypting and decrypting large quantities of information, and also permits PPEs 650 without PK-capability to deal with VDE objects 300 in a variety of applications. Using PK cryptosystems for communications may provide advantages such as eliminating

reliance on secret shared external communication keys to establish communications, allowing for a challenge/response that doesn't rely on shared internal secrets to authenticate PPEs 650, and allowing for a publicly available "certification" process without reliance on shared secret keys.

Some content providers may wish to restrict use of their content to PK implementations. This desire can be supported by making the availability of PK capabilities, and the specific nature or type of PK capabilities, in PPEs 650 a factor in the registration of VDE objects 300, for example, by including a requirement in a REGISTER method for such objects in the form of a load module that examines a PPE 650 for specific or general PK capabilities before allowing registration to continue.

Although VDE 100 does not require any specific algorithm, it is highly desirable that all PPEs 650 are capable of using the same algorithm for bulk encryption/decryption. If the bulk encryption/decryption algorithm used for encrypting VDE objects 300 is not standardized, then it is possible that not all VDE electronic appliances 600 will be capable of handling all VDE objects 300. Performance differences will exist between different PPEs 650 and associated electronic appliances 600 if standardized bulk encryption/decryption algorithms are not

implemented in whole or in part by hardware-based  
encrypt/decrypt engine 522, and instead are implemented in  
software. In order to support algorithms that are not  
implemented in whole or in part by encrypt/decrypt engine 522, a  
5 component assembly that implements such an algorithm must be  
available to a PPE 650.

### B. Key Length

Increased key length may increase security. A "brute-  
10 force" attack of a cryptosystem involves trying every possible  
key. The longer the key, the more possible keys there are to try.  
At some key length, available computation resources will require  
an impractically large amount of time for a "brute force" attacker  
to try every possible key.

15

VDE 100 provided by the preferred embodiment  
accommodates and can use many different key lengths. The  
length of keys used by VDE 100 in the preferred embodiment is  
determined by the algorithm(s) used for encryption/decryption,  
20 the level of security desired, and throughput requirements.  
Longer keys generally require additional processing power to  
ensure fast encryption/decryption response times. Therefore,  
there is a tradeoff between (a) security, and (b) processing time  
and/or resources. Since a hardware-based PPE encrypt/decrypt

engine 522 may provide faster processing than software-based encryption/decryption, the hardware-based approach may, in general, allow use of longer keys.

5           The preferred embodiment may use a 1024 bit modulus (key) RSA cryptosystem implementation for PK encryption/decryption, and may use 56-bit DES for "bulk" encryption/decryption. Since the 56-bit key provided by standard DES may not be long enough to provide sufficient security for at  
10       least the most sensitive VDE information, multiple DES encryptions using multiple passes and multiple DES keys may be used to provide additional security. DES can be made significantly more secure if operated in a manner that uses multiple passes with different keys. For example, three passes  
15       with 2 or 3 separate keys is much more secure because it effectively increases the length of the key. RC2 and RC4 (alternatives to DES) can be exported for up to 40-bit key sizes, but the key size probably needs to be much greater to provide even DES level security. The 80-bit key length provided by  
20       NSA's Skipjack may be adequate for most VDE security needs.

          The capability of downloading code and other information dynamically into PPE 650 allows key length to be adjusted and changed dynamically even after a significant number of VDE

electronic appliances 600 are in use. The ability of a VDE administrator to communicate with each PPE 650 efficiently makes such after-the-fact dynamic changes both possible and cost-effective. New or modified cryptosystems can be

5 downloaded into existing PPEs 650 to replace or add to the cryptosystem repertoire available within the PPE, allowing older PPEs to maintain compatibility with newer PPEs and/or newly released VDE objects 300 and other VDE-protected information. For example, software encryption/decryption algorithms may be

10 downloaded into PPE 650 at any time to supplement the hardware-based functionality of encrypt/decrypt engine 522 by providing different key length capabilities. To provide increased flexibility, PPE encrypt/decrypt engine 522 may be configured to anticipate multiple passes and/or variable and/or longer key

15 lengths. In addition, it may be desirable to provide PPEs 650 with the capability to internally generate longer PK keys.

### **C. Key Generation**

Key generation techniques provided by the preferred

20 embodiment permit PPE 650 to generate keys and other information that are "known" only to it.

The security of encrypted information rests in the security of the key used to encrypt it. If a cryptographically weak process

is used to generate keys, the entire security is weak. Good keys are random bit strings so that every possible key in the key space is equally likely. Therefore, keys should in general be derived from a reliably random source, for example, by a

5 cryptographically secure pseudo-random number generator seeded from such a source. Examples of such key generators are described in Schneier, Applied Cryptography (John Wiley and Sons, 1994), chapter 15. If keys are generated outside a given PPE 650 (e.g., by another PPE 650), they must be verified to

10 ensure they come from a trusted source before they can be used. "Certification" may be used to verify keys.

The preferred embodiment PPE 650 provides for the automatic generation of keys. For example, the preferred

15 embodiment PPE 650 may generate its own public/private key pair for use in protecting PK-based external communications and for other reasons. A PPE 650 may also generate its own symmetric keys for various purposes during and after initialization. Because a PPE 650 provides a secure

20 environment, most key generation in the preferred embodiment may occur within the PPE (with the possible exception of initial PPE keys used at manufacturing or installation time to allow a PPE to authenticate initial download messages to it).



Good key generation relies on randomness. The preferred embodiment PPE 650 may, as mentioned above in connection with Figure 9, include a hardware-based random number generator 542 with the characteristics required to generate reliable random numbers. These random numbers may be used to "seed" a cryptographically strong pseudo-random number generator (e.g., DES operated in Output Feedback Mode) for generation of additional key values derived from the random seed. In the preferred embodiment, random number generator 542 may consist of a "noise diode" or other physically-based source of random values (e.g., radioactive decay).

If no random number generator 542 is available in the PPE 650, the SPE 503 may employ a cryptographic algorithm (e.g., DES in Output Feedback Mode) to generate a sequence of pseudo-random values derived from a secret value protected within the SPE. Although these numbers are pseudo-random rather than truly random, they are cryptographically derived from a value unknown outside the SPE 503 and therefore may be satisfactory in some applications.

In an embodiment incorporating an HPE 655 without an SPE 503, the random value generator 565 software may derive reliably random numbers from unpredictable external physical

events (e.g., high-resolution timing of disk I/O completions or of user keystrokes at an attached keyboard 612).

Conventional techniques for generating PK and non-PK keys based upon such "seeds" may be used. Thus, if performance and manufacturing costs permit, PPE 650 in the preferred embodiment will generate its own public/private key pair based on such random or pseudo-random "seed" values. This key pair may then be used for external communications between the PPE 650 that generated the key pair and other PPEs that wish to communicate with it. For example, the generating PPE 650 may reveal the public key of the key pair to other PPEs. This allows other PPEs 650 using the public key to encrypt messages that may be decrypted only by the generating PPE (the generating PPE is the only PPE that "knows" the corresponding "private key"). Similarly, the generating PPE 650 may encrypt messages using its private key that, when decrypted successfully by other PPEs with the generating PPE's public key, permit the other PPEs to authenticate that the generating PPE sent the message.

20

Before one PPE 650 uses a public key generated by another PPE, a public key certification process should be used to provide authenticity certificates for the public key. A public-key certificate is someone's public key "signed" by a trustworthy

entity such as an authentic PPE 650 or a VDE administrator.

Certificates are used to thwart attempts to convince a PPE 650

that it is communicating with an authentic PPE when it is not

(e.g., it is actually communicating with a person attempting to

5 break the security of PPE 650). One or more VDE

administrators in the preferred embodiment may constitute a

certifying authority. By "signing" both the public key generated

by a PPE 650 and information about the PPE and/or the

corresponding VDE electronic appliance 600 (e.g., site ID, user

10 ID, expiration date, name, address, etc.), the VDE administrator

certifying authority can certify that information about the PPE

and/or the VDE electronic appliance is correct and that the

public key belongs to that particular VDE mode.

15           Certificates play an important role in the trustedness of  
digital signatures, and also are important in the public-key  
authentication communications protocol (to be discussed below).

In the preferred embodiment, these certificates may include  
information about the trustedness/level of security of a particular

20 VDE electronic appliance 600 (e.g., whether or not it has a  
hardware-based SPE 503 or is instead a less trusted software  
emulation type HPE 655) that can be used to avoid transmitting  
certain highly secure information to less trusted/secure VDE  
installations.

Certificates can also play an important role in decommissioning rogue users and/or sites. By including a site and/or user ID in a certificate, a PPE can evaluate this information as an aspect of authentication. For example, if a VDE administrator or clearinghouse encounters a certificate bearing an ID (or other information) that meets certain criteria (e.g., is present on a list of decommissioned and/or otherwise suspicious users and/or sites), they may choose to take actions based on those criteria such as refusing to communicate, communicating disabling information, notifying the user of the condition, etc. Certificates also typically include an expiration date to ensure that certificates must be replaced periodically, for example, to ensure that sites and/or users must stay in contact with a VDE administrator and/or to allow certification keys to be changed periodically. More than one certificate based on different keys may be issued for sites and/or users so that if a given certification key is compromised, one or more "backup" certificates may be used. If a certification key is compromised, A VDE administrator may refuse to authenticate based on certificates generated with such a key, and send a signal after authenticating with a "backup" certificate that invalidates all use of the compromised key and all certificates associated with it in further interactions with VDE participants. A new one or

more "backup" certificates and keys may be created and sent to the authenticated site/user after such a compromise.

5        If multiple certificates are available, some of the  
certificates may be reserved as backups. Alternatively or in  
addition, one certificate from a group of certificates may be  
selected (e.g., by using RNG 542) in a given authentication,  
thereby reducing the likelihood that a certificate associated with  
a compromised certification key will be used. Still alternatively,  
10       more than one certificate may be used in a given authentication.

      To guard against the possibility of compromise of the  
certification algorithm (e.g., by an unpredictable advance in the  
mathematical foundations on which the algorithm is based),  
15       distinct algorithms may be used for different certificates that are  
based on different mathematical foundations.

      Another technique that may be employed to decrease the  
probability of compromise is to keep secret (in protected storage  
20       in the PPE 650) the "public" values on which the certificates are  
based, thereby denying an attacker access to values that may aid  
in the attack. Although these values are nominally "public," they  
need be known only to those components which actually validate  
certificates (i.e., the PPE 650).

In the preferred embodiment, PPE 650 may generate its own certificate, or the certificate may be obtained externally, such as from a certifying authority VDE administrator.

Irrespective of where the digital certificate is generated, the  
5 certificate is eventually registered by the VDE administrator certifying authority so that other VDE electronic appliances 600 may have access to (and trust) the public key. For example, PPE 650 may communicate its public key and other information to a certifying authority which may then encrypt the public key and  
10 other information using the certifying authority's private key. Other installations 600 may trust the "certificate" because it can be authenticated by using the certifying authority's public key to decrypt it. As another example, the certifying authority may encrypt the public key it receives from the generating PPE 650  
15 and use it to encrypt the certifying authority's private key. The certifying authority may then send this encrypted information back to the generating PPE 650. The generating PPE 650 may then use the certifying authority's private key to internally create a digital certificate, after which it may destroy its copy of  
20 the certifying authority's private key. The generating PPE 650 may then send out its digital certificate to be stored in a certification repository at the VDE administrator (or elsewhere) if desired. The certificate process can also be implemented with an external key pair generator and certificate generator, but

might be somewhat less secure depending on the nature of the secure facility. In such a case, a manufacturing key should be used in PPE 650 to limit exposure to the other keys involved.

5           A PPE 650 may need more than one certificate. For example, a certificate may be needed to assure other users that a PPE is authentic, and to identify the PPE. Further certificates may be needed for individual users of a PPE 650. These certificates may incorporate both user and site information or  
10           may only include user information. Generally, a certifying authority will require a valid site certificate to be presented prior to creating a certificate for a given user. Users may each require their own public key/private key pair in order to obtain certificates. VDE administrators, clearinghouses, and other  
15           participants may normally require authentication of both the site (PPE 650) and of the user in a communication or other interaction. The processes described above for key generation and certification for PPEs 650 may also be used to form site/user certificates or user certificates.

20

Certificates as described above may also be used to certify the origin of load modules 1100 and/or the authenticity of administrative operations. The security and assurance techniques described above may be employed to decrease the

probability of compromise for any such certificate (including certificates other than the certificate for a VDE electronic appliance 600's identity).

5       **D. Key Aging and Convolution**

PPE 650 also has the ability in the preferred embodiment to generate secret keys and other information that is shared between multiple PPEs 650. In the preferred embodiment, such secret keys and other information may be shared between multiple VDE electronic appliances 600 without requiring the shared secret information to ever be communicated explicitly between the electronic appliances. More specifically, PPE 650 uses a technique called "key convolution" to derive keys based on a deterministic process in response to seed information shared between multiple VDE electronic appliances 600. Since the multiple electronic appliances 600 "know" what the "seed" information is and also "know" the deterministic process used to generate keys based on this information, each of the electronic appliances may independently generate the "true key." This permits multiple VDE electronic appliances 600 to share a common secret key without potentially compromising its security by communicating it over an insecure channel.



No encryption key should be used for an indefinite period.

The longer a key is used, the greater the chance that it may be compromised and the greater the potential loss if the key is compromised but still in use to protect new information. The longer a key is used, the more information it may protect and therefore the greater the potential rewards for someone to spend the effort necessary to break it. Further, if a key is used for a long time, there may be more ciphertext available to an attacker attempting to break the key using a ciphertext-based attack. See Schneier at 150-151. Key convolution in the preferred embodiment provides a way to efficiently change keys stored in secure database 610 on a routine periodic or other basis while simplifying key management issues surrounding the change of keys. In addition, key convolution may be used to provide "time aged keys" (discussed below) to provide "expiration dates" for key usage and/or validity.

Figure 62 shows an example implementation of key convolution in the preferred embodiment. Key convolution may be performed using a combination of a site ID 2821 and the high-order bits of the RTC 528 to yield a site-unique value "V" that is time-dependent on a large scale (e.g., hours or days). This value "V" may be used as the key for an encryption process 2871 that transforms a convolution seed value 2861 into a "current

convolution key" 2862. The seed value 2861 may be a universe-wide or group-wide shared secret value, and may be stored in secure key storage (e.g., protected memory within PPE 650). The seed value 2861 is installed during the manufacturing process  
5 and may be updated occasionally by a VDE administrator. There may be a plurality of seed values 2861 corresponding to different sets of objects 300.

The current convolution key 2862 represents an encoding  
10 of the site ID 2821 and current time. This transformed value 2862 may be used as a key for another encryption process 2872 to transform the stored key 810 in the object's PERC 808 into the true private body key 2863 for the object's contents.

15 The "convolution function" performed by blocks 2861, 2871 may, for example, be a one-way function that can be performed independently at both the content creator's site and at the content user's site. If the content user does not use precisely the same convolution function and precisely the same input values  
20 (e.g., time and/or site and/or other information) as used by the content creator, then the result of the convolution function performed by the content user will be different from the content creator's result. If the result is used as a symmetrical key for encryption by the content creator, the content user will not be

able to decrypt unless the content user's result is the same as the result of the content creator.

5       The time component for input to the key convolution  
function may be derived from RTC 528 (care being taken to  
ensure that slight differences in RTC synchronization between  
VDE electronic appliances will not cause different electronic  
appliances to use different time components). Different portions  
of the RTC 528 output may be used to provide keys with different  
10       valid durations, or some tolerance can be built into the process to  
try several different key values. For example, a "time  
granularity" parameter can be adjusted to provide time tolerance  
in terms of days, weeks, or any other time period. As one  
example, if the "time granularity" is set to 2 days, and the  
15       tolerance is  $\pm 2$  days, then three real-time input values can be  
tried as input to the convolution algorithm. Each of the resulting  
key values may be tried to determine which of the possible keys  
is actually used. In this example, the keys will have only a 4 day  
life span.

20

Figure 63 shows how an appropriate convoluted key may  
be picked in order to compensate for skew between the user's  
RTC 528 and the producer's RTC 528. A sequence of convolution  
keys 2862 (a-e) may be generated by using different input values

2881(a-e), each derived from the site ID 2821 and the RTC 528 value plus or minus a differential (e.g., -2 days, -1 days, no delta, +1 days, +2 days). The convolution steps 2871(a-e) are used to generate the sequence of keys 2862(a-e).

5

Meanwhile, the creator site may use the convolution step 2871(z) based on his RTC 528 value (adjusted to correspond to the intended validity time for the key) to generate a convoluted key 2862(z), which may then be used to generate the content key 2863 in the object's PERC 808. To decrypt the object's content, the user site may use each of its sequence of convolution keys 2862 (a-e) to attempt to generate the master content key 810. When this is attempted, as long as the RTC 538 of the creator site is within acceptable tolerance of the RTC 528 at the user site, one of keys 2862(a-e) will match key 2862(z) and the decryption will be successful. In this example, matching is determined by validity of decrypted output, not by direct comparison of keys.

20 Key convolution as described above need not use both site ID and time as a value. Some keys may be generated based on current real time, other keys might be generated on site ID, and still other keys might be generated based on both current real-time and site ID.

Key convolution can be used to provide "time-aged" keys. Such "time-aged" keys provide an automatic mechanism for allowing keys to expire and be replaced by "new" keys. They provide a way to give a user time-limited rights to make time-limited use of an object, or portions of an object, without requiring user re-registration but retaining significant control in the hands of the content provider or administrator. If secure database 610 is sufficiently secure, similar capabilities can be accomplished by checking an expiration date/time associated with a key, but this requires using more storage space for each key or group of keys.

In the preferred embodiment, PERCs 808 can include an expiration date and/or time after which access to the VDE-protected information they correspond to is no longer authorized. Alternatively or in addition, after a duration of time related to some aspect of the use of the electronic appliance 600 or one or more VDE objects 300, a PERC 808 can force a user to send audit history information to a clearinghouse, distributor, client administrator, or object creator in order to regain or retain the right to use the object(s). The PERC 808 can enforce such time-based restrictions by checking/enforcing parameters that limit key usage and/or availability past time of authorized use. "Time

aged" keys may be used to enforce or enhance this type of time-related control of access to VDE protected information.

5 "Time aged" keys can be used to encrypt and decrypt a set of information for a limited period of time, thus requiring re-registration or the receipt of new permissions or the passing of audit information, without which new keys are not provided for user use. Time aged keys can also be used to improve system security since one or more keys would be automatically replaced  
10 based on the time ageing criteria—and thus, cracking secure database 610 and locating one or more keys may have no real value. Still another advantage of using time aged keys is that they can be generated dynamically—thereby obviating the need to store decryption keys in secondary and/or secure memory.

15

A "time aged key" in the preferred embodiment is not a "true key" that can be used for encryption/decryption, but rather is a piece of information that a PPE 650, in conjunction with other information, can use to generate a "true key." This other  
20 information can be time-based, based on the particular "ID" of the PPE 650, or both. Because the "true key" is never exposed but is always generated within a secure PPE 650 environment, and because secure PPEs are required to generate the "true key,"

VDE 100 can use "time aged" keys to significantly enhance security and flexibility of the system.

5           The process of "aging" a key in the preferred embodiment involves generating a time-aged "true key" that is a function of: (a) a "true key," and (b) some other information (e.g., real time parameters, site ID parameters, etc.) This information is combined/transformed (e.g., using the "key convolution" techniques discussed above) to recover or provide a "true key."

10       Since the "true key" can be recovered, this avoids having to store the "true key" within PERC 808, and allow different "true keys" to correspond to the same information within PERC 808. Because the "true key" is not stored in the PERC 808, access to the PERC does not provide access to the information protected by

15       the "true key." Thus, "time aged" keys allows content creators/providers to impose a limitation (e.g., site based and/or time based) on information access that is, in a sense, "external of" or auxiliary to the permissioning provided by one or more PERCs 808. For example, a "time aged" key may enforce an

20       additional time limitation on access to certain protected information, this additional time limitation being independent of any information or permissioning contained within the PERC 808 and being instead based on one or more time and/or site ID values.

As one example, time-aged decryption keys may be used to allow the purchaser of a "trial subscription" of an electronically published newspaper to access each edition of the paper for a period of one week, after which the decryption keys will no longer work. In this example, the user would need to purchase one or more new PERCs 808, or receive an update to an existing one or more permissions records, to access editions other than the ones from that week. Access to those other editions which might be handled with a totally different pricing structure (e.g., a "regular" subscription rate as opposed to a free or minimal "trial" subscription rate).

In the preferred embodiment, time-aged-based "true keys" can be generated using a one-way or invertible "key convolution" function. Input parameters to the convolution function may include the supplied time-aged keys; user and/or site specific values; a specified portion (e.g., a certain number of high order bits) of the time value from an RTC 528 (if present) or a value derived from such time value in a predefined manner; and a block or record identifier that may be used to ensure that each time aged key is unique. The output of the "key convolution" function may be a "true key" that is used for decryption purposes until discarded. Running the function with a time-aged key and



inappropriate time values typically yields a useless key that will not decrypt.

5                   Generation of a new time aged key can be triggered based  
on some value of elapsed, absolute or relative time (e.g., based on  
a real time value from a clock such as RTC 528). At that time,  
the convolution would produce the wrong key and decryption  
could not occur until the time-aged key is updated. The criteria  
used to determine when a new "time aged key" is to be created  
10                   may itself be changed based on time or some other input variable  
to provide yet another level of security. Thus, the convolution  
function and/or the event invoking it may change, shift or employ  
a varying quantity as a parameter.

- 15                   One example of the use of time-aged keys is as follows:
- 1)       A creator makes a "true" key, and encrypts content  
          with it.
  - 2)       A creator performs a "reverse convolution" to yield a  
20                   "time aged key" using, as input parameters to the  
          "reverse convolution":
    - a)       the "true" key,
    - b)       a time parameter (e.g., valid high-order  
          time bits of RTC 528), and

- c) optional other information (e.g., site ID and/or user ID).

5           3) The creator distributes the "time-aged key" to content users (the creator may also need to distribute the convolution algorithm and/or parameters if she is not using a convolution algorithm already available to the content users' PPE 650).

- 10           4) The content user's PPE 650 combines:
- a) "time-aged" key
  - b) high-order time bits
  - c) required other information (same as 2c).

15           It performs a convolution function (i.e., the inverse of "reverse convolution" algorithm in step (2) above) to obtain the "true" key. If the supplied time and/or other information is "wrong," the convolution function will not yield the "true" key, and therefore content cannot be decrypted.

20           Any of the key blocks associated with VDE objects 300 or other items can be either a regular key block or a time-aged key block, as specified by the object creator during the object configuration process, or where appropriate, a distributor or client administrator.

"Time aged" keys can also be used as part of protocols to provide secure communications between PPEs 650. For example, instead of providing "true" keys to PPE 650 for communications, VDE 100 may provide only "partial" communication keys to the PPE. These "partial" keys may be provided to PPE 650 during initialization, for example. A predetermined algorithm may produce "true keys" for use to encrypt/decrypt information for secure communications. The predetermined algorithm can "age" these keys the same way in all PPEs 650, or PPEs 650 can be required to contact a VDE administrator at some predetermined time so a new set of partial communications keys can be downloaded to the PPEs. If the PPE 650 does not generate or otherwise obtain "new" partial keys, then it will be disabled from communicating with other PPEs (a further, "fail safe" key may be provided to ensure that the PPE can communicate with a VDE administrator for reinitialization purposes). Two sets of partial keys can be maintained within a PPE 650 to allow a fixed amount of overlap time across all VDE appliances 600. The older of the two sets of partial keys can be updated periodically.

20

The following additional types of keys (to be discussed below) can also be "aged" in the preferred embodiment:

individual message keys (i.e., keys used for a particular message),

administrative, stationary and travelling object shared  
keys,  
secure database keys, and  
private body and content keys.

5

### **Initial Installation Key Management**

Figure 64 shows the flow of universe-wide, or "master,"  
keys during creating of a PPE 650. In the preferred  
embodiment, the PPE 650 contains a secure non-volatile key  
storage 2802 (e.g. SPU 500 non-volatile RAM 534 B or protected  
10 storage maintained by HPE 655) that is initialized with keys  
generated by the manufacturer and by the PPE itself.

The manufacturer possesses (i.e., knows, and protects from  
15 disclosure or modification) one or more public key 2811/private  
key 2812 key pairs used for signing and validating site  
identification certificates 2821. For each site, the manufacturer  
generates a site ID 2821 and list of site characteristics 2822. In  
addition, the manufacturer possesses the public keys 2813, 2814  
20 for validating load modules and initialization code downloads.  
To enhance security, there may be a plurality of such  
certification keys, and each PPE 650 may be initialized using  
only a subset of such keys of each type.

As part of the initialization process, the PPE 650 may generate internally or the manufacturer may generate and supply, one or more pairs of site-specific public keys 2815 and private keys 2816. These are used by the PPE 650 to prove its  
5 identity. Similarly, site-specific database key(s) 2817 for the site are generated, and if needed (i.e., if a Random Number Generator 542 is not available), a random initialization seed 2818 is generated.

10 The initialization may begin by generating site ID 2821 and characteristics 2822 and the site public key 2815/private key 2816 pair(s). These values are combined and may be used to generate one or more site identity certificates 2823. The site  
15 identity certificates 2823 may be generated by the public key generation process 2804, and may be stored both in the PPE's protected key storage 2802 and in the manufacturer's VDE site certificate database 2803.

The certification process 2804 may be performed either by  
20 the manufacturer or internally to the PPE 650. If performed by the PPE 650, the PPE will temporarily receive the identity certification private key(s) 2812, generate the certificate 2823, store the certificate in local key storage 2802 and transmit it to

the manufacturer, after which the PPE 650 must erase its copy of the identity certification private key(s) 2812.

Subsequently, initialization may require generation, by  
5 the PPE 650 or by the manufacturer, of site-specific database key(s) 2817 and of site-specific seed value(s) 2818, which are stored in the key storage 2802. In addition, the download  
certification key(s) 2814 and the load module certification key(s) 2813 may be supplied by the manufacturer and stored in the key  
10 storage 2802. These may be used by the PPE 650 to validate all further communications with external entities.

At this point, the PPE 650 may be further initialized with executable code and data by downloading information certified  
15 by the load module key(s) 2813 and download key(s) 2814. In the preferred embodiment, these keys may be used to digitally sign data to be loaded into the PPE 650, guaranteeing its validity, and additional key(s) encrypted using the site-specific public  
key(s) 2815 may be used to encrypt such data and protect it from  
20 disclosure.

#### **Installation and Update Key Management**

Figure 65 illustrates an example of further key installation either by the manufacturer or by a subsequent update by a VDE

administrator. The manufacturer or administrator may supply initial or new values for private header key(s) 2831, external communication key(s) 2832, administrative object keys 2833, or other shared key(s) 2834. These keys may be universe-wide in the same sense as the global certification keys 2811, 2813, and 2814, or they may be restricted to use within a defined group of VDE instances.

To perform this installation, the installer retrieves the destination site's identity certificate(s) 2823, and from that extracts the site public key(s) 2815. These key(s) may be used in an encryption process 2841 to protect the keys being installed. The key(s) being installed are then transmitted inside the destination site's PPE 650. Inside the PPE 650, the decryption process 2842 may use the site private key(s) 2816 to decrypt the transmission. The PPE 650 then stores the installed or updated keys in its key storage 2802.

#### **Object-Specific Key Use**

Figures 66 and 67 illustrate the use of keys in protecting data and control information associated with VDE objects 300.

Figure 66 shows use of a stationary content object 850 whose control information is derived from an administrative

object 870. The objects may be received by the PPE 650 (e.g., by retrieval from an object repository 728 over a network or retrieved from local storage). The administrative object decryption process 2843 may use the private header key(s) 2815  
5 to decrypt the administrative object 870, thus retrieving the PERC 808 governing access to the content object 850. The private body key(s) 810 may then be extracted from the PERC 808 and used by the content decryption process 2845 to make the content available outside the PPE 650. In addition, the database  
10 key(s) 2817 may be used by the encryption process 2844 to prepare the PERC for storage outside the PPE 650 in the secure database 610. In subsequent access to the content object 850, the PERC 808 may be retrieved from the secure database 610, decrypted with database key(s) 2817, and used directly, rather  
15 than being extracted from administrative object 870.

Figure 67 shows the similar process involving a traveling object 860. The principal distinction between Figures 66 and 67 is that the PERC 808 is stored directly within the traveling  
20 object 860, and therefore may be used immediately after the decryption process 2843 to provide a private header key(s) 2831. This private header key 2831 is used to process content within the traveling object 860.



### **Secret-Key Variations**

Figures 64 through 67 illustrate the preferred public-key embodiment, but may also be used to help understand the secret-key versions. In secret-key embodiments, the certification process and the public key encryptions/decryptions are replaced with private-key encryptions, and the public key/private-key pairs are replaced with individual secret keys that are shared between the PPE 650 instance and the other parties (e.g., the load module supplier(s), the PPE manufacturer). In addition, the certificate generation process 2804 is not performed in secret-key embodiments, and no site identity certificates 2823 or VDE certificate database 2803 exist.

### **Key Types**

The detailed descriptions of key types below further explain secret-key embodiments; this summary is not intended as a complete description. The preferred embodiment PPE 650 can use different types of keys and/or different "shared secrets" for different purposes. Some key types apply to a Public-Key/Secret Key implementation, other keys apply to a Secret Key only implementation, and still other key types apply to both. The following table lists examples of various key and "shared secret" information used in the preferred embodiment, and where this information is used and stored:

	Key/Secret Information Type	Used in PK or Non-PK	Example Storage Location(s)
5	Master Key(s) (may include some of the specific keys mentioned below)	Both	PPE Manufacturing facility VDE administrator
	Manufacturing Key	Both (PK optional)	PPE (PK case) Manufacturing facility
	Certification key pair	PK	PPE Certification repository
	Public/private key pair	PK	PPE Certification repository (Public Key only)
10	Initial secret key	Non-PK	PPE
	PPE manufacturing ID	Non-PK	PPE
	Site ID, shared code, shared keys and shared secrets	Both	PPE
15	Download authorization key	Both	PPE VDE administrator
	External communication keys and other info	Both	PPE Secure Database
20	Administrative object keys	Both	Permission record
	Stationary object keys	Both	Permission record
	Traveling object shared keys	Both	Permission record
25	Secure database keys	Both	PPE
	Private body keys	Both	Secure database Some objects
	Content keys	Both	Secure database Some objects
	Authorization shared secrets	Both	Permission record
30	Secure Database Backup keys	Both	PPE Secure database

### Master Keys

A "master" key is a key used to encrypt other keys. An initial or "master" key may be provided within PPE 650 for

communicating other keys in a secure way. During initialization of PPE 650, code and shared keys are downloaded to the PPE. Since the code contains secure convolution algorithms and/or coefficients, it is comparable to a "master key." The shared keys  
5 may also be considered "master keys."

If public-key cryptography is used as the basis for external communication with PPE 650, then a master key is required during the PPE Public-key pair certification process. This  
10 master key may be, for example, a private key used by the manufacturer or VDE administrator to establish the digital certificate (encrypted public key and other information of the PPE), or it may, as another example, be a private key used by a VDE administrator to encrypt the entries in a certification  
15 repository. Once certification has occurred, external communications between PPEs 650 may be established using the certificates of communicating PPEs.

If shared secret keys are used as the basis for external  
20 communications, then an initial secret key is required to establish external communications for PPE 650 initialization. This initial secret key is a "master key" in the sense that it is used to encrypt other keys. A set of shared partial external communications keys (see discussion above) may be downloaded

during the PPE initialization process, and these keys are used to establish subsequent external PPE communications.

### **Manufacturing Key**

5           A manufacturing key is used at the time of PPE  
manufacture to prevent knowledge by the manufacturing staff of  
PPE-specific key information that is downloaded into a PPE at  
initialization time. For example, a PPE 650 that operates as  
part of the manufacturing facility may generate information for  
10       download into the PPE being initialized. This information must  
be encrypted during communication between the PPEs 650 to  
keep it confidential, or otherwise the manufacturing staff could  
read the information. A manufacturing key is used to protect the  
information. The manufacturing key may be used to protect  
15       various other keys downloaded into the PPE such as, for  
example, a certification private key, a PPE public/private key  
pair, and/or other keys such as shared secret keys specific to the  
PPE. Since the manufacturing key is used to encrypt other keys,  
it is a "master key."

20

A manufacturing key may be public-key based, or it may  
be based on a shared secret. Once the information is  
downloaded, the now-initialized PPE 650 can discard (or simply  
not use) the manufacturing key. A manufacturing key may be

hardwired into PPE 650 at manufacturing time, or sent to the PPE as its first key and discarded after it is no longer needed. As indicated in the table above and in the preceding discussion, a manufacturing key is not required if PK capabilities are included in the PPE.

### **Certification Key Pair**

A certification key pair may be used as part of a "certification" process for PPEs 650 and VDE electronic appliances 600. This certification process in the preferred embodiment may be used to permit a VDE electronic appliance to present one or more "certificates" authenticating that it (or its key) can be trusted. As described above, this "certification" process may be used by one PPE 650 to "certify" that it is an authentic VDE PPE, it has a certain level of security and capability set (e.g., it is hardware based rather than merely software based), etc. Briefly, the "certification" process may involve using a certificate private key of a certification key pair to encrypt a message including another VDE node's public-key. The private key of a certification key pair is preferably used to generate a PPE certificate. It is used to encrypt a public-key of the PPE. A PPE certificate can either be stored in the PPE, or it may be stored in a certification repository.

Depending on the authentication technique chosen, the public key and the private key of a certification key pair may need to be protected. In the preferred embodiment, the certification public key(s) is distributed amongst PPEs such that they may make use of them in decrypting certificates as an aspect of authentication. Since, in the preferred embodiment, this public key is used inside a PPE 650, there is no need for this public key to be available in plaintext, and in any event it is important that such key be maintained and transmitted with integrity (e.g., during initialization and/or update by a VDE administrator). If the certification public key is kept confidential (i.e., only available in plaintext inside the PPE 650), it may make cracking security much more difficult. The private key of a certification key pair should be kept confidential and only be stored by a certifying authority (i.e., should not be distributed).

In order to allow, in the preferred embodiment, the ability to differentiate installations with different levels/degrees of trustedness/security, different certification key pairs may be used (e.g., different certification keys may be used to certify SPEs 503 then are used to certify HPEs 655).

**PPE Public/Private Key Pair**

In the preferred embodiment, each PPE 650 may have its own unique "device" (and/or user) public/private key pair.

5 Preferably, the private key of this key pair is generated within the PPE and is never exposed in any form outside of the PPE.

Thus, in one embodiment, the PPE 650 may be provided with an internal capability for generating key pairs internally. If the PPE generates its own public-key crypto-system key pairs internally, a manufacturing key discussed above may not be  
10 needed. If desired, however, for cost reasons a key pair may be exposed only at the time a PPE 650 is manufactured, and may be protected at that time using a manufacturing key. Allowing PPE 650 to generate its public key pair internally allows the key pair to be concealed, but may in some applications be outweighed by  
15 the cost of putting a public-key key pair generator into PPE 650.

**Initial Secret Key**

The initial secret key is used as a master key by a secret key only based PPE 650 to protect information downloaded into  
20 the PPE during initialization. It is generated by the PPE 650, and is sent from the PPE to a secure manufacturing database encrypted using a manufacturing key. The secure database sends back a unique PPE manufacturing ID encrypted using the initial secret key in response.

The initial secret key is likely to be a much longer key than keys used for "standard" encryption due to its special role in PPE initialization. Since the resulting decryption overhead occurs only during the initialization process, multiple passes  
5 through the decryption hardware with selected portions of this key are tolerable.

### **PPE Manufacturing ID**

The PPE manufacturing ID is not a "key," but does fall  
10 within the classic definition of a "shared secret." It preferably uniquely identifies a PPE 650 and may be used by the secure database 610 to determine the PPE's initial secret key during the PPE initialization process.

### **15 Site ID, Shared Code, Shared Keys and Shared Secrets**

The VDE site ID along with shared code, keys and secrets are preferably either downloaded into PPE 650 during the PPE initialization process, or are generated internally by a PPE as part of that process. In the preferred embodiment, most or all of  
20 this information is downloaded.

The PPE site ID uniquely identifies the PPE 650. The site ID is preferably unique so as to uniquely identify the PPE 650 and distinguish that PPE from all other PPEs. The site ID in the



preferred embodiment provides a unique address that may be used for various purposes, such as for example to provide "address privacy" functions. In some cases, the site ID may be the public key of the PPE 650. In other cases, the PPE site ID  
5 may be assigned during the manufacturing and/or initialization process. In the case of a PPE 650 that is not public-key-capable, it would not be desirable to use the device secret key as the unique site ID because this would expose too many bits of the key—and therefore a different information string should be used  
10 as the site ID.

Shared code comprises those code fragments that provide at least a portion of the control program for the PPE 650. In the preferred embodiment, a basic code fragment is installed during  
15 PPE manufacturing that permits the PPE to bootstrap and begin the initialization process. This fragment can be replaced during the initialization process, or during subsequent download processing, with updated control logic.

20 Shared keys may be downloaded into PPE 650 during the initialization process. These keys may be used, for example, to decrypt the private headers of many object structures.

When PPE 650 is operating in a secret key only mode, the initialization and download processes may import shared secrets into the PPE 650. These shared secrets may be used during communications processes to permit PPEs 650 to authenticate the identity of other PPEs and/or users.

#### **Download Authorization Key**

The download authorization key is received by PPE 650 during the initialization download process. It is used to authorize further PPE 650 code updates, key updates, and may also be used to protect PPE secure database 610 backup to allow recovery by a VDE administrator (for example) if the PPE fails. It may be used along with the site ID, time and convolution algorithm to derive a site ID specific key. The download authorization key may also be used to encrypt the key block used to encrypt secure database 610 backups. It may also be used to form a site specific key that is used to enable future downloads to the PPE 650. This download authorization key is not shared among all PPEs 650 in the preferred embodiment; it is specific to functions performed by authorized VDE administrators.

### **External Communications Keys and Related Secret and Public Information**

5        There are several cases where keys are required when PPEs 650 communicate. The process of establishing secure communications may also require the use of related public and secret information about the communicating electronic appliances 600. The external communication keys and other information are used to support and authenticate secure  
10        communications. These keys comprise a public-key pair in the preferred embodiment although shared secret keys may be used alternatively or in addition.

### **Administrative Object Keys**

15        In the preferred embodiment, an administrative object shared key may be used to decrypt the private header of an administrative object 870. In the case of administrative objects, a permissions record 808 may be present in the private header. In some cases, the permissions record 808 may be distributed as  
20        (or within) an administrative object that performs the function of providing a right to process the content of other administrative objects. The permissions record 808 preferably contains the keys for the private body, and the keys for the content that can be accessed would be budgets referenced in that permissions record

808. The administrative object shared keys may incorporate time as a component, and may be replaced when expired.

### **Stationary Object Keys**

5           A stationary object shared key may be used to decrypt a private header of stationary objects 850. As explained above, in some cases a permissions record 808 may be present in the private header of stationary objects. If present, the permissions record 808 may contain the keys for the private body but will not  
10           contain the keys for the content. These shared keys may incorporate time as a component, and may be replaced when expired.

### **Traveling Object Shared Keys**

15           A traveling object shared key may be used to decrypt the private header of traveling objects 860. In the preferred embodiment, traveling objects contain permissions record 808 in their private headers. The permissions record 808 preferably contains the keys for the private body and the keys for the  
20           content that can be accessed as permitted by the permissions record 808. These shared keys may incorporate time as a component, and may be replaced when expired.

### **Secure Database Keys**

PPE 650 preferably generates these secure database keys and never exposes them outside of the PPE. They are site-specific in the preferred embodiment, and may be "aged" as described above. As described above, each time an updated record is written to secure database 610, a new key may be used and kept in a key list within the PPE. Periodically (and when the internal list has no more room), the PPE 650 may generate a new key to encrypt new or old records. A group of keys may be used instead of a single key, depending on the size of the secure database 610.

### **Private Body Keys**

Private body keys are unique to an object 300, and are not dependent on key information shared between PPEs 650. They are preferably generated by the PPE 650 at the time the private body is encrypted, and may incorporate real-time as a component to "age" them. They are received in permissions records 808, and their usage may be controlled by budgets.

### **Content Keys**

Content Keys are unique to an object 300, and are not dependent on key information shared between PPEs 650. They are preferably generated by the PPE 650 at the time the content

is encrypted. They may incorporate time as a component to "age" them. They are received in permissions records 808, and their usage may be controlled by budgets.

## 5      **Authorization Shared Secrets**

Access to and use of information within a PPE 650 or within a secure database 610 may be controlled using authorization "shared secrets" rather than keys. Authorization shared secrets may be stored within the records they authorize (permissions records 808, budget records, etc.). The authorization shared secret may be formulated when the corresponding record is created. Authorization shared secrets can be generated by an authorizing PPE 650, and may be replaced when record updates occur. Authorization shared secrets have some characteristics associated with "capabilities" used in capabilities based operating systems. Access tags (described below) are an important set of authorization shared secrets in the preferred embodiment.

## 20     **Backup Keys**

As described above, the secure database 610 backup consists of reading all secure database records and current audit "roll ups" stored in both PPE 650 and externally. Then, the backup process decrypts and re-encrypts this information using a

new set of generated keys. These keys, the time of the backup, and other appropriate information to identify the backup, may be encrypted multiple times and stored with the previously encrypted secure database files and roll up data within the backup files. These files may then all be encrypted using a "backup key" that is generated and stored within PPE 650. This backup key 500 may be used by the PPE to recover a backup if necessary. The backup keys may also be securely encrypted (e.g., using a download authentication key and/or a VDE administrator public key) and stored within the backup itself to permit a VDE administrator to recover the backup in case of PPE 650 failure.

#### Cryptographic Sealing

Sealing is used to protect the integrity of information when it may be subjected to modifications outside the control of the PPE 650, either accidentally or as an attack on the VDE security. Two specific applications may be the computation of check values for database records and the protection of data blocks that are swapped out of an SPE 500.

There are two types of sealing: keyless sealing, also known as cryptographic hashing, and keyed sealing. Both employ a cryptographically strong hash function, such as MD5 or

SHA. Such a function takes an input of arbitrary size and yields a fixed-size hash, or "digest." The digest has the property that it is infeasible to compute two inputs that yield the same digest, and infeasible to compute one input that yields a specific digest value, where "infeasible" is with reference to a work factor based on the size of the digest value in bits. If, for example, a 256-bit hash function is to be called strong, it must require approximately on average  $10^{38}$  ( $2^{128}$ ) trials before a duplicated or specified digest value is likely to be produced.

10

Keyless seals may be employed as check values in database records (e.g., in PERC 808) and similar applications. A keyless seal may be computed based on the content of the body of the record, and the seal stored with the rest of the record. The combination of seal and record may be encrypted to protect it in storage. If someone modifies the encrypted record without knowing the encryption key (either in the part representing the data or the part representing the seal), the decrypted content will be different, and the decrypted check value will not match the digest computed from the record's data. Even though the hash algorithm is known, it is not feasible to modify both the record's data and its seal to correspond because both are encrypted.

15

20



Keyed seals may be employed as protection for data stored outside a protected environment without encryption, or as a validity proof between two protected environments. A keyed seal is computed similarly to a keyless seal, except that a secret  
5 initial value is logically prefixed to the data being sealed. The digest value thus depends both on the secret and the data, and it is infeasible to compute a new seal to correspond to modified data even though the data itself is visible to an attacker. A keyed seal may protect data in storage with a single secret value,  
10 or may protect data in transit between two environments that share a single secret value.

The choice of keys or keyless seals depends on the nature of the data being protected and whether it is additionally  
15 protected by encryption.

### Tagging

Tagging is particularly useful for supporting the secure storage of important component assembly and related  
20 information on secondary storage memory 652. Integrated use of information "tagging" and encryption strategies allows use of inexpensive mass storage devices to securely store information that, in part enables, limits and/or records the configuration,

management and operation of a VDE node and the use of VDE protected content.

5           When encrypted or otherwise secured information is delivered into a user's secure VDE processing area (e.g., PPE 650), a portion of this information can be used as a "tag" that is first decrypted or otherwise unsecured and then compared to an expected value to confirm that the information represents expected information. The tag thus can be used as a portion of a  
10           process confirming the identity and correctness of received, VDE protected, information.

Three classes of tags that may be included in the control structures of the preferred embodiment:

- 15                     •       access tags
- validation tags
- correlation tags.

These tags have distinct purposes.

20           An access tag may be used as a "shared secret" between VDE protected elements and entities authorized to read and/or modify the tagged element(s). The access tag may be broken into separate fields to control different activities independently. If an access tag is used by an element such as a method core 1000',

administrative events that affect such an element must include the access tag (or portion of the access tag) for the affected element(s) and assert that tag when an event is submitted for processing. If access tags are maintained securely (e.g., created  
5 inside a PPE 650 when the elements are created, and only released from PPE 650 in encrypted structures), and only distributed to authorized parties, modification of structures can be controlled more securely. Of course, control structures (e.g., PERCs 808) may further limit or qualify modifications or other  
10 actions expressed in administrative events.

Correlation tags are used when one element references another element. For example, a creator might be required by a budget owner to obtain permission and establish a business  
15 relationship prior to referencing their budget within the creator's PERCs. After such relationship was formed, the budget owner might transmit one or more correlation tags to the creator as one aspect of allowing the creator to produce PERCs that reference the budget owner's budget.

20

Validation tags may be used to help detect record substitution attempts on the part of a tamperer.

In some respects, these three classes of tags overlap in function. For example, a correlation tag mismatch may prevent some classes of modification attempts that would normally be prevented by an access tag mismatch before an access tag check is performed. The preferred embodiment may use this overlap in some cases to reduce overhead by, for example, using access tags in a role similar to validation tags as described above.

In general, tagging procedures involve changing, within SPE 503, encryption key(s), securing techniques(s), and/or providing specific, stored tag(s). These procedures can be employed with secure database 610 information stored on said inexpensive mass storage 652 and used within a hardware SPU 500 for authenticating, decrypting, or otherwise analyzing, using and making available VDE protected content and management database information. Normally, changing validation tags involves storing within a VDE node hardware (e.g., the PPE 650) one or more elements of information corresponding to the tagging changes. Storage of information outside of the hardware SPE's physically secure, trusted environment is a highly cost savings means of secure storage, and the security of important stored management database information is enhanced by this tagging of information. Performing this tagging "change" frequently (for example, every time a given record is decrypted) prevents the

substitution of "incorrect" information for "correct" information, since said substitution will not carry information which will match the tagging information stored within the hardware SPE during subsequent retrieval of the information.

5

Another benefit of information tagging is the use of tags to help enforce and/or verify information and/or control mechanisms in force between two or more parties. If information is tagged by one party, and then passed to another party or parties, a tag can be used as an expected value associated with communications and/or transactions between the two parties regarding the tagged information. For example, if a tag is associated with a data element that is passed by Party A to Party B, Party B may require Party A to prove knowledge of the correct value of at least a portion of a tag before information related to, and/or part of, said data element is released by Party B to Party A, or vice versa. In another example, a tag may be used by Party A to verify that information sent by Party B is actually associated with, and/or part of, a tagged data element, or vice versa.

20

### **Establishing A Secure, Authenticated, Communication Channel**

From time to time, two parties (e.g., PPEs A and B), will need to establish a communication channel that is known by

both parties to be secure from eavesdropping, secure from tampering, and to be in use solely by the two parties whose identifies are correctly known to each other.

5           The following describes an example process for establishing such a channel and identifies how the requirements for security and authentication may be established and validated by the parties. The process is described in the abstract, in terms of the claims and belief each party must establish, and is not to  
10 be taken as a specification of any particular protocol. In particular, the individual sub-steps of each step are not required to be implemented using distinct operations; in practice, the establishment and validation of related proofs is often combined into a single operation.

15           The sub-steps need not be performed in the order detailed below, except to the extent that the validity of a claim cannot be proven before the claim is made by the other party. The steps may involve additional communications between the two parties  
20 than are implied by the enumerated sub-steps, as the "transmission" of information may itself be broken into sub-steps. Also, it is not necessary to protect the claims or the proofs from disclosure or modification during transmission. Knowledge of the claims (including the specific communication proposals

and acknowledgements thereof) is not considered protected information. Any modification of the proofs will cause the proofs to become invalid and will cause the process to fail.

5                   Standard public-key or secret-key cryptographic techniques can be used to implement this process (e.g., X.509, Authenticated Diffie-Hellman, Kerberos). The preferred embodiment uses the three-way X.509 public key protocol steps.

10                   The following may be the first two steps in the example process:

                  A.    (*precursor step*): Establish means of creating validatable claims by A

                  B.    (*precursor step*): Establish means of creating  
15                   validatable claims by B

                  These two steps ensure that each party has a means of making claims that can be validated by the other party, for instance, by using a public key signature scheme in which both  
20                   parties maintain a private key and make available a public key that itself is authenticated by the digital signature of a certification authority.

                  The next steps may be:

A (proposal step):

1. Determine B's identity
2. Acquire means of validating claims made by B
3. Create a unique identity for this specific proposed  
5 communication
4. Create a communication proposal identifying the  
parties and the specific communication
5. Create validatable proof of A's identity and the  
origin of the communication proposal
- 10 6. Deliver communication proposal and associated  
proof to B.

These steps establish the identity of the correspondent  
party B and proposes a communication. Because establishment  
15 of the communication will require validation of claims made by  
B, a means must be provided for A to validate such claims.  
Because the establishment of the communication must be unique  
to a specific requirement by A for communication, this  
communication proposal and all associated traffic must be  
20 unambiguously distinguishable from all other such traffic.  
Because B must validate the proposal as a legitimate proposal  
from A, a proof must be provided that the proposal is valid.



The next steps may be as follows:

B (acknowledgement step):

1. Extract A's identity from the communication proposal
- 5 2. Acquire means of validating claims made by A
3. Validate A's claim of identity and communication proposal origin
4. Determine the unique identification of the communication proposal
- 10 5. Determine that the communication proposal does not duplicate an earlier proposal
6. Create an acknowledgement identifying the specific communication proposal
- 15 7. Create validatable proof of B's identity and the origin of the acknowledgement
8. Deliver the acknowledgement and associated proof to A.

20 These steps establish that party B has received A's communication proposal and is prepared to act on it. Because B must validate the proposal, B must first determine its origin and validate its authenticity. B must ensure that its response is associated with a specific proposal, and that the proposal is not a

replay. If B accepts the proposal, it must prove both B's own identity and that B has received a specific proposal.

The next steps may be:

5

A (establishment step):

1. Validate B's claim acknowledgement of A's specific proposal
2. Extract the identity of the specific communication proposal from the acknowledgement
- 10 3. Determine that the acknowledgement is associated with an outstanding communication proposal
4. Create unique session key to be used for the proposed communication
5. Create proof of session key's creation by A
- 15 6. Create proof of session key's association with the specific communication proposal
7. Create proof of receipt of B's acknowledgement
8. Protect the session key from disclosure in transmission
- 20 9. Protect the session key from modification in transmission
10. Deliver protected session key and all proofs to B.

These steps allows A to specify a session key to be associated with all further traffic related to A's specific communication proposal. A must create the key, prove that A created it, and prove that it is associated with the specific proposed communication. In addition, A must prove that the session key is generated in response to B's acknowledgement of the proposal. The session key must be protected from disclosure of modification to ensure that an attacker cannot substitute a different value.

#### Transportability of VDE Installations Between PPEs 650

In a preferred embodiment, VDE objects 300 and other secure information may if appropriate, be transported from one PPE 650 to another securely using the various keys outlined above. VDE 100 uses redistribution of VDE administrative information to exchange ownership of VDE object 300, and to allow the portability of objects between electronic appliances 600.

The permissions record 808 of VDE objects 300 contains rights information that may be used to determine whether an object can be redistributed in whole, in part, or at all. If a VDE object 300 can be redistributed, then electronic appliance 600 normally must have a "budget" and/or other permissioning that allows it to redistribute the object. For example, an electronic

appliance 600 authorized to redistribute an object may create an administrative object containing a budget or rights less than or equal to the budget or rights that it owns. Some administrative objects may be sent to other PPEs 650. A PPE 650 that receives one of the administrative objects may have the ability to use at least a portion of the budgets, or rights, to related objects.

Transfer of ownership of a VDE object 300 is a special case in which all of the permissions and/or budgets for a VDE object are redistributed to a different PPE 650. Some VDE objects may require that all object-related information be delivered (e.g., it's possible to "sell" all rights to the object). However, some VDE objects 300 may prohibit such a transfer. In the case of ownership transfer, the original providers for a VDE object 300 may need to be contacted by the new owner, informed of the transfer, and validated using an authorization shared secret that accompanies reauthorization, before transfer of ownership can be completed.

When an electronic appliance 600 receives a component assembly, an encrypted part of the assembly may contain a value that is known only to the party or PPE 650 that supplied the assembly. This value may be saved with information that must eventually returned to the assembly supplier (e.g., audit, billing

and related information). When a component supplier requests that information be reported, the value may be provided by the supplier so that the local electronic appliance 600 can check it against the originally supplied value to ensure that the request is legitimate. When a new component is received, the value may be checked against an old component to determine whether the new component is legitimate (e.g., the new value for use in the next report process may be included with the new component).

## 10 Integrity of VDE Security

There are many ways in which a PPE 650 might be compromised. The goal of the security provided by VDE 100 is to reduce the possibility that the system will be compromised, and minimize the adverse effects if it is compromised.

15

The basic cryptographic algorithm that are used to implement VDE 100 are assumed to be safe (cryptographically strong). These include the secret-key encryption of content, public-key signatures for integrity verification, public-key encryption for privacy between PPEs 650 or between a PPE and a VDE administrator, etc. Direct attack on these algorithms is assumed to be beyond the capabilities of an attacker. For domestic versions of VDE 100 some of this is probably a safe assumption since the basic building blocks for control

information have sufficiently long keys and are sufficiently proven.

The following risks of threat or attacks may be significant:

- 5                   •     Unauthorized creation or modification of component assemblies (e.g., budgets)
- Unauthorized bulk disclosure of content
- Compromise of one or more keys
- Software emulation of a hardware PPE
- 10               •     Substitution of older records in place of newer records
- Introduction of "rogue" (i.e., unauthentic) load modules
- Replay attacks
- 15               •     Defeating "fingerprinting"
- Unauthorized disclosure of individual content items
- Redistribution of individual content items.

20               A significant potential security breach would occur if one or more encryption keys are compromised. As discussed above, however, the encryption keys used by VDE 100 are sufficiently varied and compartmentalized so that compromising one key would have only limited value to an attacker in most cases. For example, if a certification private key is exposed, an attacker

could pass the challenge/response protocol as discussed above but would then confront the next level of security that would entail cracking either the initialization challenge/response or the external communication keys. If the initialization

5 challenge/response security is also defeated, the initialization code and various initialization keys would also be exposed. However, it would still be necessary to understand the code and data to find the shared VDE keys and to duplicate the key-generation ("convolution") algorithms. In addition, correct real

10 time clock values must be maintained by the spoof. If the attacker is able to accomplish all of this successfully, then all secure communications to the bogus PPE would be compromised. An object would be compromised if communications related to the permissions record 808 of that object are sent to the bogus

15 PPE.

Knowledge of the PPE download authorization key and the algorithms that are used to derive the key that encrypts the keys for backup of secured database 610 would compromise the entire

20 secured database at a specific electronic appliance 600. However, in order to use this information to compromise content of VDE objects 300, an understanding of appropriate VDE internals would also be required. In a preferred embodiment, the private body keys and content keys stored in a secured

database 610 are "aged" by including a time component. Time is convoluted with the stored values to derive the "true keys" needed to decrypt content. If this process is also compromised, then object content or methods would be revealed. Since a  
5 backup of secured database 610 is not ever restored to a PPE 650 in the preferred embodiment without the intervention of an authorized VDE administrator, a "bogus" PPE would have to be used to make use of this information.

10 External communication shared keys are used in the preferred embodiment in conjunction with a key convolution algorithm based on site ID and time. If compromised, all of the steps necessary to allow communications with PPEs 650 must also be known to take advantage of this knowledge. In addition,  
15 at least one of the administrative object shared keys must be compromised to gain access to a decrypted permissions record  
808.

Compromising an administrative object shared key has no  
20 value unless the "cracker" also has knowledge of external communication keys. All administrative objects are encrypted by unique keys exchanged using the shared external communications keys, site ID and time. Knowledge of PPE 650



internal details would be necessary to further decrypt the content of administrative objects.

5           The private header of a stationary object (or any other stationary object that uses the same shared key) if compromised, may provide the attacker with access to content until the shared key "ages" enough to no longer decrypt the private header. Neither the private body nor the content of the object is exposed unless a permissions record 808 for that object is also  
10           compromised. The private headers of these objects may remain compromised until the key "ages" enough to no longer decrypt the private header.

15           Secure database encryption keys in the preferred embodiment are frequently changing and are also site specific. The consequences of compromising a secured database 610 file or a record depends on the information that has been compromised. For example, permissions record 808 contain keys for the public body and content of a VDE object 300. If a permissions record  
20           808 is compromised, the aspects of that object protected by the keys provided by the permissions record are also compromised—if the algorithm that generates the "true keys" is also known. If a private body key becomes known, the private body of the object is compromised until the key "ages" and

expires. If the "aging" process for that key is also compromised, the breach is permanent. Since the private body may contain methods that are shared by a number of different objects, these methods may also become compromised. When the breach is  
5 detected, all administrative objects that provide budgets and permissions record should update the compromised methods. Methods stored in secure database 610 are only replaced by more recent versions, so the compromised version becomes unusable after the update is completed.

10

If a content key becomes compromised, the portion of the content encrypted with the key is also compromised until the key "ages" and expires. If the "aging" process for that key also becomes compromised, then the breach becomes permanent. If  
15 multiple levels of encryption are used, or portions of the content are encrypted with different keys, learning a single key would be insufficient to release some or all of the content.

If an authorization shared secret (e.g., an access tag)  
20 becomes known, the record containing the secret may be modified by an authorized means if the "cracker" knows how to properly use the secret. Generally speaking, the external communications keys, the administrative object keys and the management file keys must also be "cracked" before a shared

secret is useful. Of course, any detailed knowledge of the protocols would also be required to make use of this information.

5 In the preferred embodiment, PPE 650 may detect whether or not it has become compromised. For example, by comparing information stored in an SPE 503 (e.g., summary service information) with information stored in secure database 610 and/or transmitted to a VDE participant (e.g., a VDE clearinghouse), discrepancies may become evident. If PPE 650  
10 (or a VDE administrator watching its activities or communicating with it) detects that it has been compromised, it may be updated with an initialization to use new code, keys and new encryption/decryption algorithms. This would limit exposure to VDE objects 300 that existed at the time the  
15 encryption scheme was broken. It is possible to require the PPE 650 to cease functioning after a certain period of time unless new code and key downloads occur. It is also possible to have VDE administrators force updates to occur. It is also likely that the desire to acquire a new VDE object 300 will provide an incentive  
20 for users to update their PPEs 650 at regular time intervals.

Finally, the end-to-end nature of VDE applications, in which content 108 flows in one direction, generating reports and bills 118 in the other, makes it possible to perform "back-end"

consistency checks. Such checks, performed in clearinghouses  
116, can detect patterns of use that may or do indicate fraud  
(e.g., excessive acquisition of protected content without any  
corresponding payment, usage records without corresponding  
5 billing records). The fine grain of usage reporting and the ready  
availability of usage records and reports in electronic form  
enables sophisticated fraud detection mechanisms to be built so  
that fraud-related costs can be kept to an acceptable level.

#### 10 Integrity of Software-Based PPE Security

As discussed above in connection with Figure 10, some  
applications may use a software-based protected processing  
environment 650 (such as a "host event processing environment"  
(HPE) 655) providing a software-based tamper resistant barrier  
15 674. Software-based tamper resistant barrier 674 may be  
created by software executing on a general-purpose CPU.  
Various software protection techniques may be used to construct  
and/or provide software-based tamper resistant barrier 674.

20 The risks or threat of attacks described above in  
connection with PPE 650 apply to a software-based PPE. An  
important threat to be countered with respect to a  
software-based tamper resistant barrier 674 is an attack based  
on a distributable computer program that can defeat the tamper

resistant barrier wherever the program is run. Since a software-based tamper resistant barrier 674 typically will not be as secure as a hardware-based tamper resistant barrier 502, it is useful to explore example steps and procedures a "cracker" might use to "crack" a software-based tamper resistant barrier.

Figures 67A and 67B show example "cracking" techniques a "cracker" might use to attack software-based tamper resistant barrier 674.

Referring to Figure 67A, the software used to create tamper resistant barrier 674 may be distributed, for example, on a storage medium 3370 such as a floppy diskette or optical disk (or, this software could be distributed electronically over network 108 and stored locally in a computer memory). The software distribution medium 3370 provides software (code and data) for loading into a computing device such as a general purpose personal computer 3372, for example. Personal computer 3372 may include, for example, a random access memory 3374 and a hard disk 3376.

In one example, the software distribution medium 3370 might include installation materials 3470 and operational materials 3472. The installation materials 3470 may be

executed by computer 3372 to install the operational materials 3472 onto the computer's hard disk 3376. The computer 3372 may then execute the operational materials 3472 from its hard disk 3376 to provide software-based protected processing environment 650 and associated software-based tamper resistant barrier 672.

In this example, one attack technique an attacker might use is to analyze software distribution medium 3370 (see Figure 67B, block 3352). Such analysis can take many forms.

Such analysis could be performed by a combination of one or more techniques. Such techniques include, but are not limited to, the following:

- An attacker can manually "dump" and/or disassemble listings of the data from medium 3370. This analysis is represented in Figure 67A by magnifying glass 3352A.
- An attacker can use cryptanalytic and/or key search techniques to decrypt any encrypted data from medium 3370.
- An attacker can use automated or semi-automated disassembly tools to explore the functions of programs stored on medium 3370 by studying the operation and flow

of the assembly language representation of the programs.  
This analysis is represented in Figure 67A by block 3352B.

- 5       •     An attacker can use software reverse-engineering tools to reconstruct high-level language representations of the programs on medium 3370, and study their functions.  
This analysis is represented in Figure 67A by block 3352C, producing source code 3371.
- 10       •     An attacker can use software reverse-engineering tools to create an equivalent program to the programs stored on medium 3370. As the equivalent program may be in a more convenient form, possibly in a higher-level language, it may be more amenable to analysis. This analysis is also represented in Figure 67A by block 3352C, producing source code 3371.
- 15       •     An attacker can use software debugging and/or simulation tools to follow and/or modify the dynamic execution of programs from medium 3370. This technique can be combined with any of the above static analysis techniques to study the program as it operates. This analysis is  
20       represented in Figure 67A by block 3352B.
- An attacker can use hardware-based debugging and/or simulation tools (e.g., an in-circuit emulator, or ICE) to follow and/or modify the dynamic execution of programs from medium 3370. This technique may be more effective

than the equivalent using software debugging and/or simulation tools because it has less potential effect on operation of the programs. This analysis is represented in Figure 67A by block 3352B.

5

Such analysis could provide clues and insights into the installation materials 3470, the operational materials 3472, or both.

10

Another attack technique could focus on the operational materials 3472 in the form in which they are installed on personal computer 3372. For example, one form of analysis might involve analyzing the on-disk copy of the installed software and/or associated data files installed on computer hard disk 3376 (see Figure 67B, block 3354). This analysis is represented in Figure 67A as a magnifying glass 3354B.

15

Because the installed operational materials 3472 can be executed by computer 3372, the analysis need not be limited to analyzing the static information stored on hard disk 3376, but could involve performing static and/or dynamic analysis of the executing software (see Figure 67B, blocks 3356, 3358). Any of the techniques described above could be used to analyze the operational material software 3472 to yield source code or other more interpretable form 3373A and/or a memory image 3373B.

20



The static and/or dynamic data within RAM 3374A could be similarly analyzed (see Figure 67A, magnifying glass 3354A).

5           The resulting source code 3373A and/or memory image 3373B could be carefully analyzed and reviewed (see magnifying glasses 3354D, 3354E) to obtain an understanding of both the static and dynamic structure and operation of operational materials 3272. Dynamic code analysis could involve, for example, tracing, single-stepping, data, or code break points of  
10           the executing software image, using analysis techniques such as described above. The executing software could be modified dynamically (for example, by patching) during normal operation to attempt to bypass its protection mechanisms and/or to learn more about how it operates (see Figure 67B, block 3360, and the  
15           "changes" inserted into Figure 67A memory image 3373B).

          A further attack technique in this example might involve comparing installed operational material 3472 software and data files among several different PPE 650 instances to identify  
20           important data structures, such as cryptographic keys (see "compare" block 3362A of Figure 67A; and Figure 67B, block 3362). The resulting list of differences 3362B could be carefully analyzed (see Figure 67A's magnifying glass 3362C) to obtain

important clues, using analysis techniques such as described above.

5 A further attack technique might involve comparing the memory and/or disk images of installed operational material 3472 software and data files in a single instance of PPE 650, after performing various operations using the PPE. This could serve to identify important data structures, such as cryptographic keys (see "compare" block 3362A of Figure 67A; and Figure 67B, block 3362). The resulting list of differences 10 3362B could be carefully analyzed (see Figure 67A's magnifying glass 3362C) to obtain important clues, using analysis techniques such as described above.

15 A further attack technique might involve analyzing the timing and/or order of modification to memory and/or disk images of installed operational material 3472 software and data files in a single instance of PPE 650, during the performance performing various operations using the PPE. This could serve 20 to identify important data structures, such as cryptographic keys (see "compare" block 3362A of Figure 67A; and Figure 67B, block 3362). The resulting list of differences 3362B could be carefully analyzed (see Figure 67A's magnifying glass 3362C) to obtain

important clues, using analysis techniques such as described above.

5       A further attack technique might involve duplicating one installed operational material 3472 instance by copying the programs and data from one personal computer 3372B to another personal computer 3372C or emulator (see Figure 67B, block 3364, and the "copy" arrow 3364A in Figure 67A). The duplicated PPE instance could be used in a variety of ways, such as, for example, to place an impostor PPE 650 instance on-line  
10       and/or to permit further dynamic analysis.

15       A still additional avenue of attack might involve, for example, saving the state of a PPE 650 (see Figure 67A, block 3366B) - for example, before the expenditure of credit - and restoring the state at a subsequent time (e.g., after a payment operation occurs) (see Figure 67A, arrows 3366A, 3366C, and Figure 67B, block 3366). The stored state information 3366B may also be analyzed (see Figure 67A, magnifying glass 3354F).

20

No software-only tamper resistant barrier 674 can be wholly effective against all of these threats. A sufficiently powerful dynamic analysis (such as one employing an in-circuit emulator) can lay bare all of the software-based PPE 650's

secrets. Nonetheless, various techniques described below in connection with Figure 69A and following make such an analysis extremely frustrating and time consuming - increasing the "work factor" to a point where it may become commercially unfeasible to attempt to "crack" a software-based tamper resistant barrier 674.

### PPE Initialization

Each PPE 650 needs to be initialized before it can be used. Initialization may occur at the manufacturer site, after the PPE 650 has been placed out in the field, or both. The manufacturing process for PPE 650 typically involves embedding within the PPE sufficient software that will allow the device to be more completely initialized at a later time. This manufacturing process may include, for example, testing the bootstrap loader and challenge-response software permanently stored within PPE 650, and loading the PPE's unique ID. These steps provide a basic VDE-capable PPE 650 that may be further initialized (e.g., after it has been installed within an electronic appliance 600 and placed in the field). In some cases, the manufacturing and further initialization processes may be combined to produce "VDE ready" PPEs 650. This description elaborates on the summary presented above with respect to Figures 64 and 65.

Figure 68 shows an example of steps that may be performed in accordance with one preferred embodiment to initialize a PPE 650. Some of the steps shown in this flowchart may be performed at the manufacturing site, and some may be performed remotely through contact between a VDE administrator and the PPE 650. Alternatively, all of the steps shown in the diagram may be performed at the manufacturing site, or all of the steps shown may be performed through remote communications between the PPE 500 and a VDE administrator.

If the initialization process 1370 is being performed at the manufacturer, PPE 650 may first be attached to a testbed. The manufacturing testbed may first reset the PPE 650 (e.g., with a power on clear) (Block 1372). If this reset is being performed at the manufacturer, then the PPE 650 preferably executes a special testbed bootstrap code that completely tests the PPE operation from a software standpoint and fails if something is wrong with the PPE. A secure communications exchange may then be established between the manufacturing testbed and the PPE 650 using an initial challenge-response interaction (Block 1374) that is preferably provided as part of the testbed bootstrap process. Once this secure communications has been established, the PPE 650 may report the results of the bootstrap tests it has performed to the manufacturing testbed. Assuming the PPE 650

has tested successfully, the manufacturing testbed may download new code into the PPE 650 to update its internal bootstrap code (Block 1376) so that it does not go through the testbed bootstrap process upon subsequent resets (Block 1376).

5 The manufacturing testbed may then load new firmware into the PPE internal non-volatile memory in order to provide additional standard and/or customized capabilities (Block 1378). For example, the manufacturing testbed may preload PPE 650 with the load modules appropriate for the particular manufacturing

10 lot. This step permits the PPE 500 to be customized at the factory for specific applications.

The manufacturing testbed may next load a unique device ID into PPE 650 (Block 1380). PPE 650 now carries a unique ID

15 that can be used for further interactions.

Blocks 1372-1380R typically are, in the preferred embodiment, performed at the manufacturing site. Blocks 1374 and 1382-1388 may be performed either at the manufacturing

20 site, after the PPE 650 has been deployed, or both.

To further initialize PPE 650, once a secure communications has been established between the PPE and the manufacturing testbed or a VDE administrator (Block 1374), any

required keys, tags or certificates are loaded into PPE 650 (Block 1382). For example, the manufacturing test bed may load its information into PPE 650 so the PPE may be initialized at a later time. Some of these values may be generated internally  
5 within PPE 650. The manufacturing testbed or VDE administrator may then initialize the PPE real time clock 528 to the current real time value (Block 1384). This provides a time and date reference for the PPE 650. The manufacturing testbed or the VDE administrator may next initialize the summary  
10 values maintained internally to the PPE 500 (Block 1386). If the PPE 650 is already installed as part of an electronic appliance 600, the PPE may at this point initialize its secure database 610 (Block 1388).

15 Figure 69 shows an example of program control steps performed by PPE 650 as part of a firmware download process (See Figure 68, Block 1378). The PPE download process is used to load externally provided firmware and/or data elements into the PPE. Firmware loads may take two forms: permanent loads  
20 for software that remains resident in the PPE 650; and transient loads for software that is being loaded for execution. A related process for storing into the secure database 610 is performed for elements that have been sent to a VDE electronic appliance 600.

PPE 650 automatically performs several checks to ensure that firmware being downloaded into the PPE has not been tampered with, replaced, or substituted before it was loaded. The download routine 1390 shown in the figure illustrates an example of such checks. Once the PPE 650 has received a new firmware item (Block 1392), it may check the item to ensure that it decrypts properly using the predetermined download or administrative object key (depending on the source of the element) (decision Block 1394). If the firmware decrypts properly ("yes" exits to decision Block 1394), the firmware as check valve may be calculated and compared against the check valve stored under the encryption wrapper of the firmware (decision Block 1396). If the two check summed values compare favorably ("yes" exit to decision Block 1396), then the PPE 650 may compare the public and private header identification tags associated with the firmware to ensure that the proper firmware was provided and had not been substituted (step not shown in the figure). Assuming this test also passes, the PPE 500 may calculate the digital signatures of the firmware (assuming digital signatures are supported by the PPE 650 and the firmware is "signed") and may check the calculated signature to ensure that it compares favorably to the digital signatures under the firmware encryption wrapper (Blocks 1398, 1400). If any of



these tests fail, then the download will be aborted ("fail" termination 1401).

Assuming all of the tests described above pass, then PPE  
5 650 determines whether the firmware is to be stored within the  
PPE (e.g., an internal non-volatile memory), or whether it is to  
be stored in the secure database 610 (decision Block 1402). If the  
firmware is to be stored within the PPE ("yes" exit to decision  
Block 1402), then the PPE 500 may simply store the information  
10 internally (Block 1404). If the firmware is to be stored within  
the secure database 610 ("no" exit to decision Block 1402), then  
the firmware may be tagged with a unique PPE-specific tag  
designed to prevent record substitution (Block 1406), and the  
firmware may then be encrypted using the appropriate secure  
15 database key and released to the secure database 610 (Block  
1408).

#### **Example Techniques for Forming Software-Based Tamper Resistant Barrier**

20 Various software protection techniques detailed above in  
connection with Figure 10 may provide software-based tamper  
resistant barrier 674 within a software-only and/or hybrid  
software/hardware protected processing environment 650. The  
following is an elaboration on those above-described techniques.

These software protection techniques may provide, for example, the following:

- An on-line registration process that results in the creation of a shared secret between the registry and the PPE 650 instance - used by the registry to create content and transactions that are meaningful only to that specific PPE instance.
- An installation program (that may be distinct from the PPE operational material software) that creates a customized installation of the PPE software unique to each PPE instance and/or associated electronic appliance 600.
- Camouflage protections that make it difficult to reverse engineer the PPE 650 operational materials during PPE operation.
- Integrity checks performed during PPE 650 operation (e.g., during on-line interactions with trusted servers) to detect compromise and minimize damage associated with any compromise.

In general, the software-based tamper resistant barrier 674 may establish "trust" primarily through uniqueness and complexity. In particular, uniqueness and customization complicate the ability of an attacker to:

- make multiple PPE instances with the same apparent identity;
  - make it harder for an attacker to create a software program(s) that will defeat the tamper-resistant barrier
- 5 674 of multiple PPE instances;
- make it harder for the attacker to reverse engineer (e.g., based upon encryption so that normal debugging/emulation and other software testing tools can't easily provide access); and
- 10 • make it more difficult for an attacker to compare multiple PPE instances to determine differences between them.

In addition, the overall software-based tamper resistant barrier 674 and associated PPE system is sufficiently complex so that it is difficult to tamper with a part of it without destroying other

15 aspects of its functionality (i.e., a "defense in depth").

Camouflaging techniques complicate an attacker's analysis through use of debugging/emulation or other software tools. For example, the PPE 650 may rewrite or overwrite memory locations immediately after using same to make their contents

20 unavailable for scrutiny. Similarly, the PPE 650 operational software may use hardware and/or time dependent sequences to prevent emulation. Additionally, some of the PPE 650 environment code may be self-modifying. These and other techniques make it much harder to crack an individual PPE 650

instance, and more importantly - much harder to write a program that could be used to defeat security on multiple PPE instances. Because the legitimate owner/user of a particular PPE instance may be trying to attack the security of his own system, these techniques assume that individual instances may eventually be cracked and provide additional security and safeguards that prevent (or make it more difficult) for the attacker who has cracked one PPE instance to use that information successfully in cracking other PPE instances. Specifically, these security techniques make it unlikely that an attacker who has successfully cracked one or a small number of PPE instances can write a program capable of compromising the security of any arbitrary other PPE instance, for example.

#### **Example Installation Process**

Briefly, the preferred example software-based PPE installation process provides the following security techniques:

- encrypted software distribution,
- installation customized on a unique instance and/or electronic appliance basis,
- encrypted on-disk form,
- installation tied to payment method,
- unique software and data layout, and
- identifiable copies.

Figure 69A shows one example technique for distributing the PPE 650 software. In this example, the PPE 650 software is distributed as two separate parts and/or media: the installation materials 3470, and the operational materials 3472. Installation materials 3470 may provide executable code and associated data structures for installing the operational materials 3472 onto a personal computer hard disk 3376, for example (see Figure 67A). The operational materials 3472 may provide executable code and associated data structures for providing protected processing environment 650 and associated software-based tamper resistant barrier 674.

In this example, installation materials 3470 and operational materials 3472 are each encrypted by a "deliverable preparation" process 3474 to provide encrypted installation materials 3470E and encrypted operational materials 3472E (the encrypted portions are indicated in Figure 69A, by cross-hatching). In this example, a small portion 3470C of the installation materials 3470 may be maintained in clear (unencrypted) form to provide an initial portion of the installation routine that may be executed without decryption. This plain text portion 3470C may, for example, provide an initial dialog, using an encrypted or other secure protocol with a trusted registry 3476 such as VDE administrator 200h for

example. This makes the distributed installation materials 3470  
and operational materials 3472 meaningless and unreadable to  
an attacker without additional information since the entire  
content (except for the initial dialog with the registry 3476) is  
5 unreadable.

In this example, the "deliverable preparation" process 3474  
may encrypt the installation materials 3470 and operational  
materials 3472 using one or more secret keys known to the  
10 registry 3476. Multiple versions of these installation materials  
3470 and operational materials 3472 may be distributed using  
different, secret keys so that compromise of one key exposes only  
a subset of the software distribution to unwanted disclosure.  
The only non-encrypted part of the software distribution in  
15 plaintext is that portion 3470C of installation materials 3470  
used to establish initial contact with the registry 3476.

The registry 3476 maintains a copy of the corresponding  
decryption keys within a key generation and cataloging structure  
20 3478. It provides these keys on demand during the registration  
process (e.g., using a secure key exchange protocol, for example)  
to only legitimate users authorized to set up a new protected  
processing environment 650.

Figures 69B-69C show example steps that may be performed by a installation routine 3470 to install a protected processing environment 650. In this example, upon coupling the installation materials 3470 to an electronic appliance 600 such as a personal computer 3372, the appliance begins executing the unencrypted installation materials portion 3470C. This plain text portion 3470C controls appliance 600 to contact registry 3476 and establish a registry dialog (Figure 69B, block 3470(1)). The appliance 600 and the registry 3476 use a secure key exchange protocol to exchange installation keys so that the registry may deliver the appropriate installation key to the appliance (Figure 69B, block 3470(2)). Using the provided installation key(s), the appliance 600 may decrypt and run additional portions of encrypted installation materials 3470E (Figure 69B, block 3470(3) and following). Based on this additional installation program execution, appliance 600 may decrypt and install encrypted operational materials 3472E (Figure 69B, block 3470(4)).

Rather than simply installing the operational materials 3472, in one example, installation materials 3470 makes the installation different for each PPE 650 instance. For example, the installation materials 3470 may customize the installation by:

- uniquely embedding important data into the installed software,
- uniquely encrypting the installed software,
- uniquely making random changes to the installed software,
- uniquely mating the installed software with a particular electronic appliance 600,
- providing a unique static and/or dynamic layout or other structure.

10

#### **Randomly Embedded Cryptographic Keys**

Installation routine 3470 may, for example, modify the operational materials 3472 to customize embedded locations where critical data such as cryptographic keys are stored. These keys may be embedded into the text of the operational materials 3472 at locations that vary with each installation. In this example, the registry 3476 may choose, on a random or pseudo-random basis, at least some of the operational material 3472 locations in which a particular installation routine 3470 may embed cryptographic keys or other critical data (see Figure 69B, block 3470(5)).

15

20

The installation process for the operational software may involve decrypting its distribution (which may be the same for all



end users) and modifying it to encode the specific locations where its critical data (e.g., cryptographic keys) are stored. These keys may be embedded within the text of the program at locations that vary with every installation. The distribution of unique  
5 information into the operational software 3472 can be based on a secret key known to the registry 3476. This key may be communicated by the registry 3476 during the registration dialog using a secure key exchange. The key is shared between the registry 3476 and the PPE 650 instance, and can serve both to  
10 organize the installed PPE software, and as the basis of subsequent integrity checks.

As shown in Figure 69D, the operational materials 3472 may include embedded locations 3480(a), 3480(b), 3480(c),  
15 3480(d), 3480(e), ... reserved for storing (embedding) critical information such as cryptographic keys. Each of these locations 3480 may initially store a random number string. In one example, the registry 3476 or installation routine 3470 performs a random operation 3482 to randomly select which subset of  
20 these locations 3480 is to be used by a particular instance for storing critical data. This selection list 3484 is applied as an input to an operation materials preparation step 3474a (part of the deliverable preparation operation 3474 shown in Figure 69A). The operation materials preparation step 3474a also

accepts, as an input, cryptographic keys from a secure key store 3486. In this example, the operation materials preparation step 3474a embeds the cryptographic keys provided by key store 3486 into the selected locations 3484 of operation materials 3472.

5

In accordance with one example, the random operation 3482 selects a subset that is much less than all of the possible locations 3480 - and the locations 3480 not used for storing cryptographic keys store random data instead. An attacker 10 attempting to analyze installed operational materials 3472 won't be able to tell the difference between real cryptographic keys and random number strings inserted into a place where cryptographic keys might be stored.

15 In this example, the random location selection 3484 (which is unique for each installation) may itself be encrypted by block 3488 based on an installation-unique key provided by key generation block 3490 for example. The encryption key may be securely maintained at registry 3476 so that the registry may 20 later notify the installation materials 3470 of this key - allowing the installation materials to decrypt the resulting encrypted key location block 3492 and recover listing 3484 of the subset of locations 3480 used for embedding cryptographic keys.

### **Embedded Customized Random Changes**

Referring once again to Figure 69B, the installed operational materials 3472 may be further customized for each instance by making random changes to reserved, unused portions of the operational materials (Figure 69B, block 3470(6)). An example of this is shown in Figure 69E. In this example, the operational materials 3472 include unused, embedded random data or code portions 3494. Another technique with similar effect is shown in Figure 69F. In this example, false code sections 3496 are included within reserved areas of the operational materials 3472. These false code sections 3496 add complexity, and may also be used as a electronic "fingerprint" to help trace copies. Because the false code sections 3496 are executable program code that are never executed (or if executed perform no actual functions other than confounding analysis by, for example, creating, modifying and/or destroying data that has no impact on the operation of PPE 650 but may appear to have such an impact), they can be used to confound analysis because they may be difficult for an attacker to distinguish from true code sections. In addition other false code may have the effect of disabling the execution of PPE 650 if executed. Correspondence Between Installed Software and Appliance "Signature". Another technique that may be used during the installation routine 3470 is to customize the operational materials 3472 by embedding a

"machine signature" into the operational materials to establish a correspondence between the installed software on a particular electronic appliance 600 (Figure 69C, block 3470(7)). This technique prevents a software-based PPE 650 from being transferred from one electronic appliance 600 to another (except through the use of the appropriate secure, verified backup mechanism).

For electronic appliances 600 where it is feasible to do so, the installation procedure 3470 may determine unique information about the electronic appliance 600 (e.g., a "signature" SIG in the sense of a unique value - not necessarily a "digital signature" in the cryptographic sense). Installation routine 3470 embeds the electronic appliance "signature" SIG in the installed operational materials 3472. Upon initialization, the operational materials 3472 validate the embedded signature value against the actual electronic appliance 600 signature SIG, and may refuse to start if the comparison fails.

Depending on the configuration of electronic appliance 600, the machine signature may consist, for example, of some combination of

- a hash of the ROM BIOS 658' (see Figure 69G),
- a hash of a disk defect map 3497a,
- the Ethernet (or other) network adapter 666 address,

- information written into an unused disk sector,
- information stored in a non-volatile CMOS RAM(such as used for hardware configuration data),
- 5     • information stored in non-volatile ("flash") memory (such as used for system or peripheral component "BIOS" programs) and/or
- hidden unique information placed into the root directory 3497b of the fixed disk drive 668.

10             Figure 69G shows an example of some of these appliance-specific signatures.

              In this example, machine signature information need not be particularly large. Security is provided by hiding the machine  
15     signature rather than on any other cryptographic strength, because there is no more secure mechanism for key storage to protect it. Thus, it is satisfactory for the signature to be just large enough (e.g., two bytes) that it is unlikely to be duplicated by chance.

20

              For some electronic appliances 600 where it can be determined that the technique is safe, an otherwise unused section of the non-volatile CMOS RAM 656a may be used to store a signature 3497d. Signature 3497d is verified against the PPE

650's internal state whenever the PPE is initialized. Signature 3497d may also be updated whenever a significant change is made to the secure database 610. If the CMOS RAM signature 3497d does not match the database value, PPE 650 may take this mismatch as an indication that a previous instance of the secure database 610 and/or PPE 650 software has been restored, and appropriate action can be taken. This mechanism thus ensures that even a bit-for-bit copy of the system's fixed disk 668 or other storage medium cannot be saved and reloaded to restore an earlier PPE 650 state. This particular technique depends upon there being an unused location available within CMOS RAM 656a, and may also require the CMOS RAM checksum algorithm to be known. An incorrect implementation could cause a subsequent reboot of electronic appliance 600 to fail because of a bad CMOS checksum, or worse, could alter some critical configuration parameter within CMOS RAM 656a so that electronic appliance 600 could not be recovered. Thus, care must be taken before modifying the contents of CMOS RAM 656a.

20           A still alternate technique may involve marking otherwise "good" disk sectors 3497c defective and using the sector(s) to store machine signatures and/or encryption keys. This technique ensures that a logical bit-for-bit copy of the media does not result in a usable PPE 650 instance, and also provides relatively

inaccessible and non-volatile storage for the information.

Because a relatively large amount of storage space can be reserved using this technique, there is enough storage for a cryptographically strong value.

5

Some of the "machine signature" techniques discussed above may be problematic in some electronic appliances 600 because it may be difficult to locate appropriate appliance-unique information. For example, although in a personal computer a ROM BIOS 658' is always available, the ROM BIOS information by itself may be insufficient because it is likely to be identical for a batch of electronic appliances 600 purchased together. Identifying a network adapter 666 and determining its address is potentially difficult due to the wide variety of adapters; additionally, an electronic appliance's network address may change (although this occurrence may be infrequent). Inserting random signature values into unused bytes within the fixed disk root directory 3497b and/or partition records may trigger some virus-checking programs, and the data may be modified by defragmentation or other disk manipulation programs. Where supported, a truly unused disk sector 3497c (e.g., one that is marked "bad" even though it may still viably store information) may be used to store the machine signature. Even so, normal maintenance, upgrades or other failure recovery

10

15

20

procedures may disrupt a particular machine association. Since the VDE administrator 200h participates in restoring a PPE 650 based on an encrypted backup image (as described above for example in connection with Figures 39-40), the VDE administrator may establish new associations at this point to maintain correspondence between a particular PPE 650 installation and a particular electronic appliance 600.

#### **Tie Installation To Payment Method**

A still additional example technique for providing additional security is to tie a particular PPE 650 installation at registration time to a particular payment method (see Figure 69C, block 3470(8)). The registration process at installation time may thus serve to tie the PPE 650 installation to some payment method associated with the user, and to store the payment association information both within the PPE 650 instance and at the registry 3476. This technique assures that the actions of a particular PPE 650 instance are accountable to the assigned user with at least the reliability of whatever payment/credit verification technique is employed.

#### **Install Operational Materials In Encrypted Form**

Operational materials 3472 may first be customized as described above for the particular instance and/or appliance 600,



then (at least mostly) encrypted for installation into the appliance such as by storage onto disk 668 (see Figure 69C, block 3470(9)). Different installations may use different sets of decryption keys to decrypt the information once installed.

5 Different parts of operational materials 3472 may be encrypted with different cryptographic keys to further complicate the analysis. This encryption makes analysis of the on disk form of the operational materials 3472 more difficult or infeasible.

10 The beginning of the resulting stored executable file may contain a small decryption program ("decryptor") that decrypts the remainder of the operational materials 3472 as they are loaded into memory. Confounding algorithms (as described below) may be used in this decryptor to make static recovery of  
15 the cryptographic keys difficult. Although the decryptor is necessarily in unencrypted form in an all-software installation without hardware support, the use of confounding algorithms to develop the associated cryptographic keys effectively requires a memory image to be captured after the program has been  
20 decrypted. Where supported (as described above), an unused and inaccessible disk sector 3497c may be used to store the decryption keys, and the operational materials 3472 may possess only the address for that particular sector. Embedding this address further complicates analysis.

### Customized Layout

5       The installation materials 3470 may store the encrypted  
operational materials 3472 onto the fixed disk 668 using a  
customized storage layout (Figure 69C, block 3470(10)). Figure  
69F, 69H, 69I and 69J shows example customized software and  
data layouts. In these examples, each installed instance of  
operational materials 3472 is different in both executable form  
and in data layout. These modifications make each PPE 650  
10       instance require separate analysis in order to determine the  
storage locations of its critical data such as cryptographic keys.  
This technique is an effective counter to creation of programs  
that can undo the protections of an arbitrary PPE 650 instance.

15       Instruction sequences within the operational materials  
3472 may be modified by the installation routine to change the  
execution flow of the executable operational materials 3472 and  
to alter the locations at which the software expects to locate  
critical data. The alterations in program flow may include  
20       customization of time-consuming confounding algorithms. The  
locations of the modifiable instruction sequences may be  
embedded within operational materials 3470, and may therefore  
be not directly available from an examination of the installation  
and/or operational materials.

Figure 69H shows one example operational materials 3472 executable code segment provided distinct processes 3498a, 3498b, 3498c, 3498d, 3498e. In this particular example, segment 3498a is executed first and segment 3498e is executed last, but the processes 3498b, 3498c and 3498d may be performed in any order (i.e., they are sequence independent processes). The installation materials 3470 may take advantage of this sequence independence by storing and/or executing them in different and/or depending upon the particular PPE instance 650. Figure 69I, for example, shows a first static layout order, and Figure 69J shows a second, different static layout order. Data elements associated with the executables may similarly be stored in different orders (as shown in Figures 69I, 69J) depending upon the particular installation.

### **Dynamic Protection Mechanisms**

In addition to the more static protection mechanisms described above, dynamic protection mechanisms may be employed to complicate both static and dynamic analysis of the executable (executing) operational materials 3472. Such techniques include, for example:

- implementation complexity,
- immediate overwriting,
- hardware dependent sequences,

- timing dependencies,
- confounding algorithms,
- random modifications,
- dynamic load module decryption,
- 5 • on-line integrity checks,
- time integrity checks,
- machine association integrity checks,
- dynamic storage integrity checks, and
- hidden secret storage
- 10 • volatile secret storage
- internal consistency checks.

Figures 69K-69L show an example execution of  
operational materials 3472 that may employ some or all of these  
various dynamic protection mechanisms.

Upon starting execution (Figure 69K, block 3550), the installed  
operational materials 3472 may run initialization code as  
described above that is used to decrypt the stored encrypted  
operational materials on an "as needed" basis (Figure 69K, block  
20 3552). This initialization code may also check the current value  
of the real-time clock (Figure 69K, block 3554).

### Real Time Check/Validation

Operational materials 3472 may perform this time check, for example, to guard against replay attacks and to ensure that the electronic appliance 600's time is in reasonable agreement  
5 with that of the VDE administrator 200h or other trusted node.

Figure 69M shows an example sequence of steps that may be performed by the "check time" block 3554. In this example, PPE 650 uses secure communications (e.g. a cryptographic  
10 protocol) to obtain the current real time from a trusted server (Figure 69M, block 3554a). PPE 650 may next ask the user if he or she wishes to reset the electronic appliance real-time clock 528 (which may, for example, be the real-time clock module within a personal computer or the like) so it is synchronized with the trusted server's time clock.

15

If the user responds affirmatively, PPE 650 may reset the time clock to agree with the real-time provided by the trusted server ("yes" exit to decision block 3554b, Figure 69M, block 3554c). If the user responds that he or she does not want the  
20 real-time clock reset ("no" exit to decision block 3554b), then PPE 650 may calculate a delta value of the difference between the server's real-time clock and the electronic appliance's real-time clock 528 (Figure 69M, block 3554d). In either case, PPE 650 may store the current time  $T_{current}$  into a non-volatile storage

location Tstore indicating the current real-time (Figure 69M, block 3554e).

5 Referring again to Figure 69K, PPE 650 can disable itself if there is too much (or the wrong type) of a difference between the trusted server's time and the electronic appliance's clock - since such differences can indicate replay attacks, the possibility that the PPE 650 has been restored based on a previous state, etc. For example, if desired, PPE 650 can generate a time check  
10 fail exception if the electronic appliance's real-time clock 528 disagrees with the trusted server's real-time by more than a certain amount of acceptable drift (Figure 69K, "yes" exit to decision block 3556). In the event of such an exception, PPE 650 may disable itself (Figure 69K, block 3558) and require a dialog  
15 between the user and registry 3476 (or other authority) - providing additional protection against replay attacks and also detecting clock failures that could lead to incorrect operation or incorrect charges.

## 20 **Dynamic Code Decryption and Data OverWriting**

Operational materials 3472 may then decrypt the next program segment dynamically (Figure 69K, block 3460. The code may be decrypted dynamically when it is needed, then re-encrypted or overwritten and discarded when not in use.

This mechanism increases the tamper-resistance of the executable code - thus providing additional tamper resistance for PPE operations. As mentioned above, different decryption keys may be required to decode different code portions, and the decryption keys can be installation-specific so that an attacker who successfully compromises the decryption key of one instance cannot use that information to compromise any other instance's decryption key(s).

Once a portion of the operational materials 3472 has been decrypted (Figure 69K, block 3560), that portion may immediately overwrite all initialization code in memory since it is no longer required (Figure 69K, block 3562). The executing operational materials 3472 may similarly overwrite all unwrapped cryptographic keys once they are no longer needed, and may also overwrite expanded key information developed by initializing the cryptographic algorithms once no longer needed. These techniques minimize the amount of time during which usable key information is available for exposure in a memory snapshot -- complicating all but the most dynamic of analysis efforts. Because all keys in permanent storage are either encrypted or otherwise camouflaged, no such treatment is required for I/O buffers.

**Dynamic Check of Association Between Appliance and PPE****Instance**

5           The executing operational materials 3472 may next  
compare an embedded electronic appliance signature SIG'  
against the electronic appliance signature SIG stored in the  
electronic appliance itself (Figure 69K, decision block 3564). As  
discussed above, this technique may be used to help prevent  
operational materials 3472 from operating on any electronic  
appliance 600 other than the one it was initially installed on.  
10       PPE 650 may disable operation if this machine signature check  
fails ("no" exit to decision block 3564, Figure 69K; disable block  
3566).

**Self-Modifying and/or Hardware-Dependent Code Sequences**

15           Executing operational materials 3472 may also employ  
self-modifying code sequences that cannot easily be emulated  
with a software debugger or single-stepping program (Figure  
~~69K~~, block 3568). These sequences may, for example, be  
dependent on specific models of electronic appliances 600, and  
20       may be patched into the operational materials 3472 as  
appropriate to installation materials 3470 based on tests  
performed during the installation process. Such  
hardware-dependent sequences may be used to ensure that  
critical algorithms yield different results when executed on the



proper hardware as opposed to when executed on different hardware or under software control such as in a debugger or emulator. To prevent such hardware-dependent sequences from being readily recognizable from a static examination of the code, the sequences may be constructed at run time and then invoked - so that they can be identified only by analysis of the instruction sequences actually executed.

### Dynamic Timing Checks

Executing operational materials 3472 may also make dynamic timing checks on various code sequences, and refuse to operate if they do not execute within the expected interval (Figure 69K, block 3570, decision block 3572, "disable" block 3574). . An incorrect execution time suggests that the operational materials 3472 are being externally manipulated and/or analyzed or traced in some manner (e.g., by a software emulator). This technique thus provides additional protection against dynamic analysis and/or modification

The expected execution intervals associated with certain code sequences may be calculated during the installation procedure. Resulting test values may be embedded into the operational materials 3472. These timing tests may be integrated with time integrity tests and dynamic integrity checks

to make it more difficult to bypass them simply by patching out the timing check. Care should be taken to eliminate false alarms due to concurrent system activity (e.g., other tasks and/or windows)..

5

Figure 69N shows one example of a dynamic time check routine 3570. In this example, a test may be performed to determine whether it is time to perform another time check (decision block 3570a). For example, this test 3570a may be performed periodically and/or at the end of a time-dependent sequence as described above. If performed periodically, a counter V value may be incremented or reset to zero - readying this counter for the next performance of test 3470A (see Figure 69N, block 3570b, 3570c, 3570d).

15

If it is time to perform a check, PPE 650 compares the stored time value  $T_{store}$  with the current time value  $T_{current}$ , and determines whether the two values are within an acceptable range (Figure 69N, decision block 3570E). If the two values agree within an acceptable range (this range may be determined, for example, in part by the time-dependent testing described above), then PPE 650 may replace the stored time value  $T_{store}$  with the current  $T_{current}$  in preparation for the next test (Figure 69N, block 3570F). If, on the other hand, the two values

20

are not within an acceptable range (the "not within range" exit to decision block 3570E, Figure 69N), PPE 650 may disable operation (block 3570G) and initiate a conversation with a trusted time base or other verification facility to perform further authenticity checking (Figure 69N, block 3570H).

As Figure 69L shows, further time checks may be performed periodically and/or repeatedly based on other events (see block 3582, decision block 3584, disable block 3586).

## 10 **Confounding Algorithms**

The executing operational materials 3472 may also perform various confounding algorithms - computationally intensive algorithms that perform a complex operation in order to generate values required at run time (Figure 69L, block 3576).

15 The purpose of such confounding algorithms is to make infrequently invoked steps (e.g., initialization or other steps not performed very frequently) inscrutable to an attacker who is disassembling or tracing them. Confounding algorithms may also be used for the time-dependent checking described above.

20 One example of such a "confounding algorithm" is a modified version of the MD5 message digest function (applied repeatedly to the same input value), which tests internally generated results of the round functions and terminates when a specific value is encountered. For example, one may make

random modifications to the confounding algorithm (for example, by adjusting the "magic constants" in MD5) until it terminates quickly enough to be useful with the desired value in some register. This adjustment may be performed beforehand to yield  
5 a prior knowledge of modifications that can then be installed differently and to each PPE 650 instance.

As one specific example, a family of 256 customized confounding algorithms could be created, each defined by a  
10 single modification of the MD5 "magic constants" (or even the input data to MD5) so that the algorithm terminates with any of 256 possible values in some register. Critical values can then be generated at run time by installing appropriate versions of the algorithm into the operational materials 3472 and assembling  
15 the values a byte at a time. Confounding algorithms may be performed in a time-dependent value as described above; their execution times may be logged and checked by PPE 650, and the PPE 650 may disable operation if the confounding algorithms run too rapidly or slowly.

20

Such confounding algorithms are generally infeasible to simulate by hand because they may require tens or hundreds of millions of instructions to complete. They are expensive to analyze at run time because single-stepping through the code is

time consuming (though not prohibitive, particularly if break points are set at all the possible termination tests rather than for every instruction). Although such confounding algorithms are expensive in computation time, then need not be invoked frequently - preserving efficiency.

#### **Random Modifications to Environment State**

The executing operational materials 3472 may randomly modify the PPE 650 environment state during normal operation to reflect both actual PPE 650 operations being performed and to include random modifications of data not significant to the operating PPE 650 (Figure 69L, block 3578). Such techniques help ensure that snapshots of the secure database 610 and operational materials 3472 cannot readily be compared to identify significant values and objects.

Such modifications may be based, for example, on actual random values derived from unpredictable hardware events such as disk I/O completion timing and keyboard timing. Such techniques make it infeasible to experiment with "minor" changes to the PPE 650 state even if the attacker can successfully bypass integrity checks that prevent duplicates from being made.

### **Load Module Dynamic Decryption & Re-Encryption**

5       The executing operational materials 3472 may decrypt  
load module 1100 code dynamically as needed, and re-encrypt it  
or otherwise render it inscrutable when not in use (Figure 69L,  
block 3580). In accordance with this technique, load module  
executable code and/or data is decrypted dynamically when it is  
needed, then re-encrypted or destroyed when not in use. In  
addition, the location of executing load modules 1100 may be  
varied randomly to foil attempts to set break points within the  
10       load module. Different algorithms and a changing key may be  
used to further confound dynamic analysis.

### **Hidden Secret Storage**

15       The source database 610 and/or parts or all of operational  
materials 3472 may be protected by cryptography employing  
keys and/or authentication values hidden in normally  
inaccessible locations in the appliance 600. If the key or  
authentication value is not available, the decryption cannot be  
performed, rendering PPE 650 unusable. Examples of such  
20       locations include, but are not limited to:

- Disk storage artificially marked as damaged (for the  
purpose of storing secrets);
- Disk storage normally reserved as alternates for sectors  
that may be marked as damaged;

- Disk storage normally reserved for non-general purpose use, such as sectors reserved by the manufacturer for firmware storage, for storage of statistics, or for test purposes, etc.;
- 5 • Non-volatile, writable, storage in the appliance or its components, such as that used for configuration data, device and controlled firmware, standard BIOS software, etc.;
- 10 • Unused storage in files maintained by an operating system, such as the bytes between the logical end of a file and the end of its last physical sector, etc.;
- Unused storage in file system control structures, such as the bytes available to store as-yet-undefined attributes. unused storage in file allocation maps and other  
15 structures, unused storage in redundant (duplicate) file maps and directories, unused or unneeded bytes in boot records, etc.;

20 Storing secrets (e.g., cryptographic keys and authentication values) in these locations serves two purposes: it makes them difficult to locate by analysis of the PPE 650, and it makes them difficult to copy between one instance of the PPE and another (or to replace the PPE's contents with an earlier version of the same).

### Volatile Secret Storage

5       The secure database 610 and/or parts or all of operational materials 3472 may be protected by cryptography employing keys and/or authentication values ("cryptovariables") that are maintained only in volatile storage during normal operation. For example, during an initialization sequence, cryptovariables can be read from permanent storage (e.g., disk), overwritten, and held only in volatile memory during system operation. During the shutdown sequence, the cryptovariables can be rewritten to permanent storage.

15       This provides resistance to tampering because the initialization sequence for an appliance, particularly a general-purpose computer, is typically more difficult to tamper with than is the computer during normal operation. This technique prevents the computer from itself being used to analyze the contents of permanent storage media; only by removing the media and analyzing it independently can the cryptovariables be located and extracted. This technique has the drawback of requiring the appliance's operation always to be terminated normally so that the termination sequence is guaranteed to update the permanent storage. This drawback can be ameliorated by maintaining frequent backups of the secure database 610 and/or the protected crypto-variables that can be



restored with administration by VDE administrator 200h if a disorderly termination occurs.

### **Dynamic Integrity Checks**

5           In this example, operational materials 3472 may also perform a variety of dynamic integrity checks that tie an executing PPE 650 to a particular electronic appliance 600 and to guard against various forms of replay or substitution attacks. One example of a replay attack, for example, is an attack in  
10       which a user restores the PPE 650 state from an earlier backup - wiping out all recent billing records. PPE 650 includes a backup mechanism (as discussed above in connection with Figures 39 and 40) that supports restoration of previous states after system failure. Executing operational materials 3472 in this example  
15       provides certain dynamic protection mechanisms (integrity checks) that prevent such backup and restoration processes from being misused to allow such replay attacks. Such checks may identify incomplete or erroneous attempts to subvert tamper resistant barrier 672. Great care must be taken to ensure that  
20       these checks do not trigger as a result of execution or implementation error, as there is potential for significant disruption.

For example, during PPE 650 operation, the internal state of the PPE is constantly being updated. During each interaction with a trusted server, PPE 650 (and the trusted server) may test the internal state of PPE 650 to determine whether it could be derived from the internal state last seen by the trusted server for this particular PPE 650 instance. If it could not, the result may be taken as indicating a replay attack of some sort, and an appropriate action can be taken (see Figure 69L, block 3592, 3594, 3596).

For example, such a check could be implemented using a counter stored in PPE 650 and updated every time an operation is performed. If the trusted server finds the counter to be smaller than at the previous server interaction, this finding is strong evidence that a previous state of the PPE 650 environment has been restored. In practice, the check might be implemented with an obscure technique to prevent easy manipulation of the counter value. For example, the counter could be repeated hashing (e.g., with MD5) of a value that is stored redundantly in several different locations within the operational materials 3472 and secure database 610 - so that the trusted server could verify that the current value can be derived (e.g., by repeated MD5 applications) from a previous value. Such checks may limit the severity of loss resulting from off-line

manipulation of PPE 650. Because the trusted server verifies the consistency of PPE 650 at each interaction, the only loss that may occur as a result of wholesale reloading of an earlier PPE 650 state is that of content that has already been delivered by  
5 has not yet been charged for.

One example of a dynamic integrity check that executing operational materials 3472 may perform (Figure 69L, block 3588) might, for example, be the periodic verification of the integrity of  
10 the operational materials code in memory by a checksum invoked by a timer. If the timer does not tick regularly, the PPE 650 may detect it and cease to operate (see Figure 69N). This verification may counter attacks that might, for example, attempt to trick PPE 650 access methods into releasing content that has been  
15 decrypted but not electronically fingerprinted. Executing operational materials 3472 may also include numerous internal consistency checks to prevent substitution (replay) of stale database 610 records, introduction of invalid load modules 1100, external modification of the secure database 610, and so on.  
20 Such checks may be made sufficiently complex and interwoven as to make modifications likely to be detected.

When an inconsistency is detected ("yes" exit to decision block 3590, Figure 69L), PPE 650 can take appropriate action

such as locking itself up from further use until reconstructed under the trusted server's control (Figure 69L, disable block 3591). For example, PPE 650 could encrypt its secure database 610 with a new, random key, then encrypt that with the server's public key. Only the server could then arrange to reconstruct the user's instance of PPE 650.

### Defense In Depth

Finally, although not a "camouflage" technique per se, the complexity of operational materials 3472 may make it difficult to understand them from the outside in. As discussed above, PPE 650 may make extensive use of RPC and coordinated work in different threads of execution. Because much of the RPC traffic may be encrypted, it will be difficult to unravel even if operational materials 3472 are heavily instrumented by the attacker. Although the cryptographic keys are, in principal, readily available in memory (e.g., because after all, the PPE 650 must be able to get them), there may be many keys and it will be difficult to identify the right one rapidly. In addition, a primary benefit to be sought by subverting protection of software-based PPE 650 installations is the ability to acquire content without paying for it - in other words, the ability to "create money". The integrity checks discussed above mean that any error in manipulating the budget and usage information data is likely to

be detected quickly. Even if the checks occur off-line without notification to any trusted server, it will make the user's PPE 650 instance effectively useless - requiring its destruction and recreation.

5

# **Networking SPU's 500 and/or VDE Electronic Appliances 600**

In the context of many computers interconnected by a local or wide area network, it would be possible for one or a few of them to be VDE electronic appliances 600. For example, a VDE- 10 capable server might include one or more SPU's 500. This centralized VDE server could provide all VDE services required within the network or it can share VDE service with VDE server nodes; that is, it can perform a few, some, or most VDE service activities. For example, a user's non-VDE computer could issue a 15 request over the network for VDE-protected content. In response to the request, the VDE server could comply by accessing the appropriate VDE object 300, releasing the requested content and delivering the content over the network 672 to the requesting user. Such an arrangement would allow VDE capabilities to be 20 easily integrated into existing networks without requiring modification or replacement of the various computers and other devices connected to the networks.

For example, a VDE server having one or more protected processing environments 650 could communicate over a network with workstations that do not have a protected processing environment. The VDE server could perform all secure VDE processing, and release resulting content and other information to the workstations on the network. This arrangement would require no hardware or software modification to the workstations.

However, some applications may require greater security, flexibility and/or performance that may be obtained by providing multiple VDE electronic appliances 600 connected to the same network 672. Because commonly-used local area networks constitute an insecure channel that may be subject to tampering and/or eavesdropping, it is desirable in most secure applications to protect the information communicated across the network. It would be possible to use conventional network security techniques to protect VDE-released content or other VDE information communicated across a network 672 between a VDE electronic appliance 600 and a non-VDE electronic appliance. However, advantages are obtained by providing multiple networked VDE electronic appliances 600 within the same system.

As discussed above in connection with Figure 8, multiple VDE electronic appliances 600 may communicate with one another over a network 672 or other communications path. Such networking of VDE electronic appliances 600 can provide advantages. Advantages include, for example, the possibility of centralizing VDE resources, storing and/or archiving metering information on a server VDE and delivering information and services efficiently across the network 672 to multiple electronic appliances 600.

For example, in a local area network topology, a "VDE server" electronic appliance 600 could store VDE-protected information and make it available to one or more additional electronic appliances 600 or computers that may communicate with the server over network 672. As one example, an object repository 728 storing VDE objects could be maintained at the centralized server, and each of many networked electronic appliance 600 users could access the centralized object repository over the network 672 as needed. When a user needs to access a particular VDE object 300, her electronic appliance 600 could issue a request over network 672 to obtain a copy of the object. The "VDE server" could deliver all or a portion of the requested object 300 in response to the request. Providing such a centralized object repository 728 would have the advantage of

minimizing mass storage requirements local to each electronic appliance 600 connected to the network 672, eliminate redundant copies of the same information, ease information management burdens, provide additional physical and/or other security for particularly important VDE processes and/or information occurring at the server, where providing such security at VDE nodes may be commercially impractical for certain business models, etc.

10           It may also be desirable to centralize secure database 610 in a local area network topology. For example, in the context of a local area network, a secure database 610 server could be provided at a centralized location. Each of several electronic appliances 600 connected to a local area network 672 could issue requests for secure database 610 records over the network, and receive those records via the network. The records could be provided over the network in encrypted form. "Keys" needed to decrypt the records could be shared by transmitting them across the network in secure communication exchanges. Centralizing secure database 610 in a network 672 has potential advantages of minimizing or eliminating secondary storage and/or other memory requirements for each of the networked electronic appliances 600, avoiding redundant information storage,



allowing centralized backup services to be provided, easing information management burdens, etc.

One way to inexpensively and conveniently deploy multiple instances of VDE electronic appliances 600 across a network would be to provide network workstations with software defining an HPE 655. This arrangement requires no hardware modification of the workstations; an HPE 655 can be defined using software only. An SPE(s) 503 and/or HPE(s) 655 could also be provided within a VDE server. This arrangement has the advantage of allowing distributed VDE network processing without requiring workstations to be customized or modified (except for loading a new program(s) into them). VDE functions requiring high levels of security may be restricted to an SPU-based VDE server. "Secure" HPE-based workstations could perform VDE functions requiring less security, and could also coordinate their activities with the VDE server.

Thus, it may be advantageous to provide multiple VDE electronic appliances 600 within the same network. It may also be advantageous to provide multiple VDE electronic appliances 600 within the same workstation or other electronic appliance 600. For example, an electronic appliance 600 may include

multiple electronic appliances 600 each of which have a SPU 500 and are capable of performing VDE functions.

For example, one or more VDE electronic appliances 600  
5 can be used as input/output device(s) of a computer system. This may eliminate the need to decrypt information in one device and then move it in unencrypted form across some bus or other unsecured channel to another device such as a peripheral. If the peripheral device itself is a VDE electronic appliance 600 having  
10 a SPU 500, VDE-protected information may be securely sent to the peripheral across the insecure channel for processing (e.g., decryption) at the peripheral device. Giving the peripheral device the capability of handling VDE-protected information directly also increases flexibility. For example, the VDE  
15 electronic appliance 600 peripheral device may control VDE object 300 usage. It may, for example, meter the usage or other parameters associated with the information it processes, and it may gather audit trails and other information specific to the processing it performs in order to provide greater information  
20 gathering about VDE object usage. Providing multiple cooperating VDE electronic appliances 600 may also increase performance by eliminating the need to move encrypted information to a VDE electronic appliance 600 and then move it again in unencrypted form to a non-VDE device. The VDE-

protected information can be moved directly to its destination device which, if VDE-capable, may directly process it without requiring involvement by some other VDE electronic appliance 600.

5

Figure 70 shows an example of an arrangement 2630 comprising multiple VDE electronic appliances 600(1), 600(2), 600(3), . . . , 600(N). VDE electronic appliances 600(1) . . . 600(N) may communicate with one another over a communications path 2631 (e.g., the system bus of a work station, a telephone or other wire, a cable, a backplane, a network 672, or any other communications mechanism). Each of the electronic appliances 600 shown in the figure may have the same general architecture shown in Figure 8, i.e., they may each include a CPU (or microcontroller) 654, SPU 500, RAM 656, ROM 658, and system bus 653. Each of the electronic appliances 600 shown in the figure may have an interface/controller 2632 (which may be considered to be a particular kind of I/O controller 660 and/or communications controller 666 shown in Figure 8). This interface/controller 2632 provides an interface between the electronic appliance system bus 653 and an appropriate electrical connector 2634. Electrical connectors 2634 of each of the respective electronic appliances 600(1), . . . 600(N) provide a

10

15

20

connection to a common network 672 or other communication paths.

5           Although each of electronic appliances 600 shown in the figure may have a generally similar architecture, they may perform different specialized tasks. For example, electronic appliance 600(1) might comprise a central processing section of a workstation responsible for managing the overall operation of the workstation and providing computation resources. Electronic  
10       appliance 600(2) might be a mass storage device 620 for the same workstation, and could provide a storage mechanism 2636 that might, for example, read information from and write information to a secondary storage device 652. Electronic appliance 600(3)  
15       might be a display device 614 responsible for performing display tasks, and could provide a displaying mechanism 2638 such as a graphics controller and associated video or other display. Electronic appliance 600(N) might be a printer 622 that performs printing related tasks and could include, for example, a print  
20       mechanism 2640.

Each of electronic appliances 600(1), . . . 600(N) could comprise a different module of the same workstation device all contained within a common housing, or the different electronic appliances could be located within different system components.

For example, electronic appliance 600(2) could be disposed within a disk controller unit, electronic appliance 600(3) could be disposed within a display device 614 housing, and the electronic appliance 600(N) could be disposed within the housing of a printer 622. Referring back to Figure 7, scanner 626, modem 618, telecommunication means 624, keyboard 612 and/or voice recognition box 613 could each comprise a VDE electronic appliance 600 having its own SPU 500. Additional examples include RF or otherwise wireless interface controller, a serial interface controller, LAN controllers, MPEG (video) controllers, etc.

Because electronic appliances 600(1) . . . 600(N) are each VDE-capable, they each have the ability to perform encryption and/or decryption of VDE-protected information. This means that information communicated across network 672 or other communications path 2631 connecting the electronic appliances can be VDE-protected (e.g., it may be packaged in the form of VDE administrative and/or content objects and encrypted as discussed above). One of the consequences of this arrangement is that an eavesdropper who taps into communications path 2631 will not be able obtain information except in VDE-protected form. For example, information generated by electronic appliance 600 (1) to be printed could be packaged in a VDE

content object 300 and transmitted over path 2631 to electronic appliance 600 (N) for printing. An attacker would gain little benefit from intercepting this information since it is transmitted in protected form; she would have to compromise electronic  
5 appliance 600(1) or 600(N) (or the SPU 500(1), 500(N)) in order to access this information in unprotected form.

Another advantage provided by the arrangement shown in the diagram is that each of electronic appliances 600(1), . . .  
10 600(N) may perform their own metering, control and/or other VDE-related functions. For example, electronic appliance 600(N) may meter and/or perform any other VDE control functions related to the information to be printed, electronic appliance 600(3) may meter and/or perform any other VDE control  
15 functions related to the information to be displayed, electronic appliance 600(2) may meter and/or perform any other VDE control functions related to the information to be stored and/or retrieved from mass storage 620, and electronic appliance 600(1) may meter and/or perform any other VDE control functions  
20 related to the information it processes.

In one specific arrangement, each of electronic appliances 600(1), . . . 600(N) would receive a command that indicates that the information received by or sent to the electronic appliance is

to use its SPU 500 to process the information to follow. For example, electronic appliance 600(N) might receive a command that indicates that information it is about to receive for printing is in VDE-protected form (or the information that is sent to it may itself indicate this). Upon receiving this command or other information, electronic appliance 600(N) may decrypt the received information using SPU 500, and might also meter the information the SPU provides to the print mechanism 2644 for printing. An additional command might be sent to electronic appliance 600(N) to disable the decryption process or 600(N)'s VDE secure subsystem may determine that the information should not be decrypted and/or printed. Additional commands, for example, may exist to load encryption/decryption keys, load "limits," establish "fingerprinting" requirements, and read metered usage. These additional commands may be sent in encrypted or unencrypted form as appropriate.

Suppose, for example, that electronic appliance 600(1) produces information it wishes to have printed by a VDE-capable printer 622. SPU 500(1) could establish a secure communications across path 2631 with SPU 500(N) to provide a command instructing SPU 500(N) to decrypt the next block of data and store it as a decryption key and a limit. SPU 500(1) might then send a further command to SPU 500(N) to use the

decryption key and associated limit to process any following encrypted print stream (or this command could be sent by CPU 654(1) to microcontroller 654(N)). Electronic appliance 600(1) could then begin sending encrypted information on path 672 for decryption and printing by printer 622. Upon receipt of each new block of information by printer 622, SPU 500(N) might first check to ensure that the limit is greater than zero. SPU 500(N) could then increment a usage meter value it maintains, and decrement the limit value. If the limit value is non-zero, SPU 500(N) could decrypt the information it has received and provide it to print mechanism 2640 for printing. If the limit is zero, then SPU 500(N) would not send the received information to the print mechanism 2640, nor would it decrypt it. Upon receipt of a command to stop, printer 622 could revert to a "non-secure" mode in which it would print everything received by it across path 2631 without permitting VDE processing.

The SPU 500(N) associated with printer 622 need not necessarily be disposed within the housing of the printer, but could instead be placed within an I/O controller 660 for example (see Figure 8). This would allow at least some of the advantages similar to the ones discussed above to be provided without requiring a special VDE-capable printer 622. Alternatively, a SPU 500(N) could be provided both within printer 622 and



within I/O controller 660 communicating with the printer to provide advantages in terms of coordinating I/O control and relieving processing burdens from the SPU 500 associated with the central processing electronic appliance 600(1). When

5 multiple VDE instances occur within an electronic appliance, one or more VDE secure subsystems may be "central" subsystems, that is "secondary" VDE instances may pass encrypted usage related information to one or more central secure subsystems so as to allow said central subsystem to directly control storage of

10 said usage related information. Certain control information may also be centrally stored by a central subsystem and all or a portion of such information may be securely provided to the secondary secure subsystem upon its secure VDE request.

Such printer protections as described above may be

15 particularly useful, for example, in the case of content providers that want to restrict or otherwise control (e.g., charge for) printing of their content. These controls can be easily enforced in the case of printers as described above having SPUs 500 (PPEs 650), but may be difficult to enforce in the case of

20 general-purpose printers that do not have an SPU (PPE). It may be relatively easy in such environments to use printer redirectors, print output to a file, or otherwise manipulate the system's printing functions. It therefore may be advantageous to

provide a strategy that protects printed outputs in such general purpose printing environments.

So-called "intelligent" printers are capable of executing "scripts" or other programs comprising executable instructions or

5 commands. Such "script" languages can be used to provide a degree of tamper-resistance and security without the necessity of an SPU 500 (PPE 650).

For example, it is possible to create a decryption program  
10 3900, in PostScript or another printer control language, that can be downloaded  
3801 to an associated printer 3901, creating a program 3904 stored inside  
15 the memory of printer 3901. Because PostScript, as well as other similar printer control languages, is a general-purpose programming language,  
such a program could decrypt (3802) an encrypted data stream  
20 3902 sent to such a printer 3901. This approach would allow a local PPE 650 to prepare (3800) files for printing inside the PPE by creating an encrypted

file 3902 to be printed and delivering it for processing by the decryption program inside the printer.

5       The decryption program 3900 could be downloaded (3801) as a printer initialization activity. Using features of the PostScript or other language, and/or other mechanisms in the printer, the decryption program 3904 could be locked into the memory 3901m of printer 3901 so it  
10       could not be viewed and/or modified except with appropriate authorization. The downloaded decryption program 3904 could engage in an interactive secure  
15       protocol dialogue (3802) with PPE 650 to demonstrate that it has not been tampered with, as a precondition to creating and delivering the encrypted printable content 3902. The decryption program 3900 could also  
20       be downloaded (3801) to printer 3901 prior to printing one or more encrypted content streams 3902. The decryption program 3904 could destroy itself

after printing one or more encrypted content streams 3902 to protect itself against viewing or tampering.

5           As shown in Figure 70B, the decryption program 3900 could alternatively or additionally provide a fingerprinting function--for example by selecting characters for printing from several related character fonts (3910a-3910z) in a pattern 3911 that is generated from a fingerprint key 3912 using standard  
10 cryptographic and/or steganographic techniques. Because each of the characters 3913aa, 3913ba, etc. representing the letter "A" in fonts 3910a-3910z is printed with a slightly different image, it is possible to identify the font from which each character was drawn by careful examination of the printed output, and thus to  
15 reconstruct the pattern 3911 and from that the fingerprint key 3912. There are similarly different patterns for other characters in fonts 3910a-3910z, shown as 3913ab-3913zb, etc., permitting a more efficient and/or tamper-resistant encoding of fingerprint information.

20

For printers that use non-executable control languages, such as PCL5, scrambled fonts can be used without requiring that a decryption program be downloaded or otherwise installed in the printer. As

shown in figure 70, it is possible download permuted font images 3921 to such printers. Scrambled fonts 3921 are created by rearranging the character images in a normal font 3920. Such fonts allow PPE 650 to scramble the data before it is delivered for printing, so that it could not be easily interpreted except by being printed on a printer with appropriate scrambled fonts. As a simple substitution cipher, this could be inverted using automated techniques, but it provides good security against accidental disclosure. Plural scrambled fonts 3921, scrambled according to different patterns, could be resident simultaneously in the printer, and used for different sections of the printed content or different pages. Scrambled fonts 3921 could be downloaded and/or replaced dynamically in the printer differently for each page or other division in the output.

The technique of using multiple character images for fingerprinting shown in figure 70B is also applicable to printers incorporating non-executable control languages, by downloading multiple fonts 3910a-z incorporating different images for characters.

### Portable Electronic Appliance

Electronic appliance 600 provided by the present invention may be portable. Figure 71 shows one example of a portable electronic appliance 2600. Portable appliance 2600 may include  
5 a portable housing 2602 that may be about the size of a credit card in one example. Housing 2602 may connect to the outside world through, for example, an electrical connector 2604 having one or more electrical contact pins (not shown). Connector 2604 may electrically connect an external bus interface 2606 internal  
10 to housing 2602 to a mating connector 2604a of a host system 2608. External bus interface 2606 may, for example, comprise a PCMCIA (or other standard) bus interface to allow portable appliance 2600 to interface with and communicate over a bus 2607 of host system 2608. Host 2608 may, for example, be  
15 almost any device imaginable, such as a computer, a pay telephone, another VDE electronic appliance 600, a television, an arcade video game, or a washing machine, to name a few examples.

20 Housing 2602 may be tamper resistant. (See discussion above relating to tamper resistance of SPU barrier 502.)

Portable appliance 2600 in the preferred embodiment includes one or more SPUs 500 that may be disposed within

housing 2602. SPU 500 may be connected to external bus interface 2606 by a bus 2610 internal to housing 2602. SPU 500 communicates with host 2608 (through external bus interface 2606) over this internal bus 2610.

5

SPU 500 may be powered by a battery 2612 or other portable power supply that is preferably disposed within housing 2602. Battery 2612 may be, for example, a miniature battery of the type found in watches or credit card sized calculators.

10 Battery 2612 may be supplemented (or replaced) by solar cells, rechargeable batteries, capacitive storage cells, etc.

A random access memory (RAM) 2614 is preferably provided within housing 2602. RAM 2614 may be connected to SPU 500 and not directly connected to bus 2610, so that the contents of RAM 2614 may be accessed only by the SPU and not by host 2608 (except through and as permitted by the SPU). Looking at Figure 9 for a moment, RAM 2614 may be part of RAM 534 within the SPU 500, although it need not necessarily be contained within the same integrated circuit or other package that houses the rest of the SPU.

20

Portable appliance 2600 RAM 534 may contain, for example, information which can be used to uniquely identify

each instance of the portable appliance. This information may be employed (e.g. as at least a portion of key or password information) in authentication, verification, decryption, and/or encryption processes.

5

Portable appliance 2600 may, in one embodiment, comprise means to perform substantially all of the functions of a VDE electronic appliance 600. Thus, for example, portable appliance 2600 may include the means for storing and using permissions, methods, keys, programs, and/or other information, and can be capable of operating as a "stand alone" VDE node.

10

In a further embodiment, portable appliance 2600 may perform preferred embodiment VDE functions once it has been coupled to an additional external electronic appliance 600. Certain information, such as database management permission(s), method(s), key(s), and/or other important information (such as at least a portion of other VDE programs: administrative, user-interface, analysis, etc.) may be stored (for example as records) at an external VDE electronic appliance 600 that may share information with portable appliance 2600.

15

20

One possible "stand alone" configuration for tamper-resistant, portable appliance 2600 arrangements



includes a tamper-resistant package (housing 2602) containing one or more processors (500, 2616) and/or other computing devices and/or other control logic, along with random-access-memory 2614. Processors 500, 2616 may execute permissions and methods wholly (or at least in part) within the portable appliance 2600. The portable appliance 2600 may have the ability to encrypt information before the information is communicated outside of the housing 2602 and/or decrypt received information when said received information is received from outside of the housing. This version would also possess the ability to store at least a portion of permission, method, and/or key information securely within said tamper resistant portable housing 2602 on non-volatile memory.

Another version of portable appliance 2600 may obtain permissions and/or methods and/or keys from a local VDE electronic appliance 600 external to the portable appliance 2600 to control, limit, or otherwise manage a user's use of a VDE protected object. Such a portable appliance 600 may be contained within, received by, installed in, or directly connected to, another electronic appliance 2600.

One example of a "minimal" configuration of portable appliance 2600 would include only SPU 500 and battery 2612

within housing 2602 (the external bus interface 2606 and the RAM 2614 would in this case each be incorporated into the SPU block shown in the Figure). In other, enhanced examples of portable appliance 2600, any or all of the following optional components may also be included within housing 2602:

one or more CPUs 2616 (with associated support components such as RAM-ROM 2617, I/O controllers (not shown), etc.);

one or more display devices 2618;

one or more keypads or other user input buttons/control information 2620;

one or more removable/replaceable memory device(s) 2622;

and

one or more printing device(s) 2624.

In such more enhanced versions, the display 2618, keypad 2620, memory device 2622 and printer 2624 may be connected to bus 2610, or they might be connected to CPU 2616 through an I/O port/controller portion (not shown) of the CPU. Display 2618 may be used to display information from SPU 500, CPU 2616 and/or host 2608. Keypad 2620 may be used to input information to SPU 500, CPU 2616 and/or host 2608. Printer 2624 may be used to print information from any/all of these sources. Removable/replaceable memory 2622 may comprise a

memory cartridge or memory medium such as a bulk storage device, for providing additional long-term or short-term storage. Memory 2622 may be easily removable from housing 2602 if desired.

5

In one example embodiment, portable appliance 2600 may have the form factor of a "smart card" (although a "smart card" form factor may provide certain advantages, housing 2602 may have the same or different form factor as "conventional" smart cards). Alternatively, such a portable electronic appliance 2600 may, for example, be packaged in a PCMCIA card configuration (or the like) which is currently becoming quite popular on personal computers and is predicted to become common for desk-top computing devices and Personal Digital Assistants.

10 One advantageous form factor for the portable electronic appliance housing 2602 may be, for example, a Type 1, 2, or 3 PCMCIA card (or other derivations) having credit card or somewhat larger dimensions. Such a form factor is conveniently portable, and may be insertable into a wide array of computers and consumer appliances, as well as receptacles at commercial establishments such as retail establishments and banks, and at public communications points, such as telephone or other telecommunication "booths."

15  
20

Housing 2602 may be insertable into and removable from a port, slot or other receptacle provided by host 2608 so as to be physically (or otherwise operatively) connected to a computer or other electronic appliance. The portable appliance connector 2604 may be configured to allow easy removability so that appliance 2600 may be moved to another computer or other electronic appliance at a different location for a physical connection or other operative connection with that other device.

Portable electronic appliance 2600 may provide a valuable and relatively simple means for a user to move permissions and methods between their (compatible) various electronic appliances 600, such as between a notebook computer, a desktop computer and an office computer. It could also be used, for example, to allow a consumer to visit a next door neighbor and allow that neighbor to watch a movie that the consumer had acquired a license to view, or perhaps to listen to an audio record on a large capacity optical disk that the consumer had licensed for unlimited plays.

Portable electronic appliance 2600 may also serve as a "smart card" for financial and other transactions for users to employ in a variety of other applications such as, for example, commercial applications. The portable electronic appliance 2600

may, for example, carry permission and/or method information used to authorize (and possibly record) commercial processes and services.

5           An advantage of using the preferred embodiment VDE portable appliance 2600 for financial transactions such as those typically performed by banks and credit card companies is that VDE allows financial clearinghouses (such as VISA, MasterCard, or American Express) to experience significant reductions in  
10           operating costs. The clearinghouse reduction in costs result from the fact that the local metering and budget management that occurs at the user site through the use of a VDE electronic appliance 600 such as portable appliance 2600 frees the clearinghouse from being involved in every transaction. In  
15           contrast to current requirements, clearinghouses will be able to perform their functions by periodically updating their records (such as once a month). Audit and/or budget "roll-ups" may occur during a connection initiated to communicate such audit and/or budget information and/or through a connection that can  
20           occur at periodic or relatively periodic intervals and/or during a credit updating, purchasing, or other portable appliance 2600 transaction.

Clearinghouse VDE digital distribution transactions would require only occasional authorization and/or audit or other administrative "roll-ups" to the central service, rather than far more costly connections during each session. Since there would be no requirement for the maintenance of a credit card purchase "paper trail" (the authorization and then forwarding of the credit card slip), there could be substantial cost reductions for clearinghouses (and, potentially, lower costs to users) due to reduction in communication costs, facilities to handle concurrent processing of information, and paper handling aspects of transaction processing costs. This use of a portable appliance 2600 would allow credit enforcement to exploit distributed processing employing the computing capability in each VDE electronic appliance 600. These credit cost and processing advantages may also apply to the use of non-smart card and non-portable VDE electronic appliance 600s.

— Since VDE 100 may be configured as a highly secure commercial environment, and since the authentication processes supported by VDE employ digital signature processes which provide a legal validation that should be equivalent to paper documentation and handwritten signatures, the need for portable appliance 2600 to maintain paper trails, even for more costly transactions, is eliminated. Since auditable billing and

control mechanisms are built into VDE 100 and automated, they may replace traditional electronic interfaces to VISA, Master Card, AMEX, and bank debit accounts for digitally distributed other products and services, and may save substantial operating costs for such clearinghouses.

Portable appliance 2600 may, if desired, maintain for a consumer a portable electronic history. The portable history can be, for example, moved to an electronic "dock" or other receptacle, in or operatively connected to, a computer or other consumer host appliance 2608. Host appliance 2608 could be, for example, an electronic organizer that has control logic at least in part in the form of a microcomputer and that stores information in an organized manner, e.g., according to tax and/or other transaction categories (such as type of use or activity). By use of this arrangement, the consumer no longer has to maintain receipts or otherwise manually track transactions but nevertheless can maintain an electronic, highly secure audit trail of transactions and transaction descriptions. The transaction descriptions may, for example, securely include the user's digital signature, and optionally, the service or goods provider's digital signature.

When a portable appliance 2600 is "docked" to a host 2608 such as a personal computer or other electronic appliance (such

as an electronic organizer), the portable appliance 2600 could communicate interim audit information to the host. In one embodiment, this information could be read, directly or indirectly, into a computer or electronic organizer money and/or tax management program (for example, Quicken or Microsoft Money and/or Turbo Tax and/or Andrew Tobias' Managing Your Money). This automation of receipt management would be an enormous boon to consumers, since the management and maintenance of receipts is difficult and time-consuming, receipts are often lost or forgotten, and the detail from credit card billings is often wholly inadequate for billing and reimbursement purposes since credit card billings normally don't provide sufficient data on the purchased items or significant transaction parameters.

15

In one embodiment, the portable appliance 2600 could support secure (in this instance encrypted and/or authenticated) two-way communications with a retail terminal which may contain a VDE electronic appliance 600 or communicate with a retailer's or third party provider's VDE electronic appliance 600. During such a secure two-way communication between, for example, each participant's secure VDE subsystem, portable appliance 2600 VDE secure subsystem may provide authentication and appropriate credit or debit card information

20



to the retail terminal VDE secure subsystem. During the same or different communication session, the terminal could similarly, securely communicate back to the portable appliance 2600 VDE secure subsystem details as to the retail transaction (for  
5 example, what was purchased and price, the retail establishment's digital signature, the retail terminal's identifier, tax related information, etc.).

For example, a host 2608 receptacle for receiving and/or  
10 attaching to portable appliance 2600 could be incorporated into or operatively connected to, a retail or other commercial establishment terminal. The host terminal 2608 could be operated by either a commercial establishment employee or by the portable appliance 2600 holder. It could be used to, for  
15 example, input specific keyboard and/or voice input specific information such as who was taken to dinner, why something was purchased, or the category that the information should be attached to. Information could then be automatically "parsed" and routed into securely maintained (for example, encrypted)  
20 appropriate database management records within portable appliance 2600. Said "parsing" and routing would be securely controlled by VDE secure subsystem processes and could, for example, be based on category information entered in by the user and/or based on class of establishment and/or type (category) of

expenditure information (or other use). Categorization can be provided by the retail establishment, for example, by securely communicating electronic category information as a portion, for example, of electronic receipt information or alternatively by printing a hard copy receipt using printer 2624. This process of categorization may take place in the portable appliance 2600 or, alternatively, it could be performed by the retail establishment and periodically "rolled-up" and communicated to the portable appliance 2600 holder.

10

Retail, clearinghouse, or other commercial organizations may maintain and use by securely communicating to appliance 2600 one or more of generic classifications of transaction types (for example, as specified by government taxation rules) that can be used to automate the parsing of information into records and/or for database information "roll-ups" for; and/or in portable appliance 2600 or one or more associated VDE nodes. In such instances, host 2608 may comprise an auxiliary terminal, for example, or it could comprise or be incorporated directly within a commercial establishments cash registers or other retail transactions devices. The auxiliary terminal could be menu and/or icon driven, and allow very easy user selection of categorization. It could also provide templates, based on transaction type, that could guide the user through specifying

20

useful or required transaction specific information (for example, purpose for a business dinner and/or who attended the dinner). For example, a user might select a business icon, then select from travel, sales, meals, administration, or purchasing icons for example, and then might enter in very specific information and/or a key word, or other code that might cause the downloading of a transaction's detail into the portable appliance 2600. This information might also be stored by the commercial establishment, and might also be communicated to the appropriate government and/or business organizations for validation of the reported transactions (the high level of security of auditing and communications and authentication and validation of VDE should be sufficiently trusted so as not to require the maintenance of a parallel audit history, but parallel maintenance may be supported, and maintained at least for a limited period of time so as to provide backup information in the event of loss or "failure" of portable appliance 2600 and/or one or more appliance 2600 associated VDE installations employed by appliance 2600 for historical and/or status information record maintenance). For example, of a retail terminal maintained necessary transaction information concerning a transaction involving appliance 2600, it could communicate such information to a clearinghouse for archiving (and/or other action) or it could periodically, for example, at the end of a business day, securely

communicate such information, for example, in the form of a VDE content container object, to a clearinghouse or clearinghouse agent. Such transaction history (and any required VDE related status information such as available credit) can be  
5 maintained and if necessary, employed to reconstruct the information in a portable appliance 2600 so as to allow a replacement appliance to be provided to an appliance 2600 user or properly reset internal information in data wherein such replacement and/or resetting provides all necessary transaction  
10 and status information.

In a retail establishment, the auxiliary terminal host 2608 might take the form of a portable device presented to the user, for example at the end of a meal. The user might place his  
15 portable appliance 2600 into a smart card receptacle such as a PCMCIA slot, and then enter whatever additional information that might appropriately describe the transaction as well as satisfying whatever electronic appliance 600 identification procedure(s) required. The transaction, given the availability of  
20 sufficient credit, would be approved, and transaction related information would then be communicated back from the auxiliary terminal directly into the portable appliance 2600. This would be a highly convenient mode of credit usage and record management.

The portable device auxiliary terminal might be "on-line," that is electronically communicating back to a commercial establishment and/or third party information collection point through the use of cellular, satellite, radio frequency, or other communications means. The auxiliary terminal might, after a check by a commercial party in response to receipt of certain identification information at the collection point, communicate back to the auxiliary terminal whether or not to accept the portable appliance 2600 based on other information, such as a bad credit record or a stolen portable appliance 2600. Such a portable auxiliary terminal would also be very useful at other commercial establishments, for example at gasoline stations, rental car return areas, street and stadium vendors, bars, and other commercial establishments where efficiency would be optimized by allowing clerks and other personnel to consummate transactions at points other than traditional cash register locations.

As mentioned above, portable appliance 2600 may communicate from time to time with other electronic appliances 600 such as, for example, a VDE administrator. Communication during a portable appliance 2600 usage session may result from internally stored parameters dictating that the connection should take place during that current session (or next or other

session) of use of the portable appliance. The portable appliance 600 can carry information concerning a real-time date or window of time or duration of time that will, when appropriate, require the communication to take place (e.g., perhaps before the transaction or other process which has been contemplated by the user for that session or during it or immediately following it). Such a communication can be accomplished quickly, and could be a secure, VDE two-way communication during which information is communicated to a central information handler. Certain other information may be communicated to the portable appliance 2600 and/or the computer or other electronic appliance to which the portable appliance 2600 has been connected. Such communicated other information can enable or prevent a contemplated process from proceeding, and/or make the portable appliance 2600, at least in part, unusable or useable. Information communicated to the portable appliance 2600 could include one or more modifications to permissions and methods, such as a resetting or increasing of one or more budgets, adding or withdrawing certain permissions, etc.

20

The permissions and/or methods (i.e., budgets) carried by the portable appliance 2600 may have been assigned to it in conjunction with an "encumbering" of another, stationary or other portable VDE electronic appliance 600. In one example, a

portable appliance 2600 holder or other VDE electronic appliance 600 and/or VDE electronic appliance 600 user could act as "guarantor" of the financial aspects of a transaction performed by another party. The portable appliance 2600 of the holder would  
5 record an "encumbrance," which may be, during a secure communication with a clearinghouse, be recorded and maintained by the clearinghouse and/or some other financial services party until all or a portion of debt responsibilities of the other party were paid or otherwise satisfied. Alternatively or in  
10 addition, the encumbrance may also be maintained within the portable appliance 2600, representing the contingent obligation of the guarantor. The encumbrance may be, by some formula, included in a determination of the credit available to the guarantor. The credit transfer, acceptance, and/or record  
15 management, and related processes, may be securely maintained by the security features provided by aspects of the present invention. Portable appliance 600 may be the sole location for said permissions and/or methods for one or more VDE objects 300, or it may carry budgets for said objects that are independent  
20 of budgets for said objects that are found on another, non-portable VDE electronic appliance 600. This may allow budgets, for example, to be portable, without requiring "encumbering" and budget reconciliation.

Portable VDE electronic appliance 2600 may carry (as may other VDE electronic appliance 600s described) information describing credit history details, summary of authorizations, and usage history information (e.g., audit of some degree of transaction history or related summary information such as the use of a certain type/class of information) that allows re-use of certain VDE protected information at no cost or at a reduced cost. Such usage or cost of usage may be contingent, at least in part, on previous use of one or more objects or class of objects or amount of use, etc., of VDE protected information.

Portable appliance 2600 may also carry certain information which may be used, at least in part, for identification purposes. This information may be employed in a certain order (e.g. a pattern such as, for example, based on a pseudo-random algorithm) to verify the identity of the carrier of the portable appliance 2600. Such information may include, for example, one's own or a wife's and/or other relatives maiden names, social security number or numbers of one's own and/or others, birth dates, birth hospital(s), and other identifying information. It may also or alternatively provide or include one or more passwords or other information used to identify or otherwise verify/authenticate an individual's identity, such as voice print and retinal scan information. For example, a portable



appliance 2600 can be used as a smart card that carries various permissions and/or method information for authorizations and budgets. This information can be stored securely within portable appliance 2600 in a secure database 610 arrangement. When a user attempts to purchase or license an electronic product or otherwise use the "smart card" to authorize a process, portable appliance 2600 may query the user for identification information or may initiate an identification process employing scanned or otherwise entered information (such as user fingerprint, retinal or voice analysis or other techniques that may, for example, employ mapping and/or matching of provided characteristics to information securely stored within the portable appliance 2600). The portable appliance 2600 may employ different queries at different times (and/or may present a plurality of queries or requests for scanning or otherwise entering identifying information) so as to prevent an individual who has come into possession of appropriate information for one or more of the "tests" of identity from being able to successfully employ the portable appliance 2600.

20

A portable appliance 600 could also have the ability to transfer electronic currency or credit to another portable appliance 2600 or to another individual's account, for example, using secure VDE communication of relevant content between

secure VDE subsystems. Such transfer may be accomplished, for example, by telecommunication to, or presentation at, a bank which can transfer credit and/or currency to the other account. The transfer could also occur by using two cards at the same

5 portable appliance 2600 docking station. For example, a credit transaction workstation could include dual PCMCIA slots and appropriate credit and/or currency transfer application software which allows securely debiting one portable appliance 2600 and "crediting" another portable appliance (i.e., debiting from one

10 appliance can occur upon issuing a corresponding credit and/or currency to the other appliance). One portable appliance 600, for example, could provide an authenticated credit to another user. Employing two "smart card" portable appliance 600 would enable the user of the providing of "credit" "smart card" to go through a

15 transaction process in which said user provides proper identification (for example, a password) and identifies a "public key" identifying another "smart card" portable appliance 2600.

The other portable appliance 2600 could use acceptance processes, and provide proper identification for a digital

20 signature (and the credit and/or currency sender may also digitally sign a transaction certificate so the sending act may not be repudiated and this certificate may accompany the credit and/or currency as VDE container content. The transactions may involve, for example, user interface interaction that

stipulates interest and/or other terms of the transfer. It may employ templates for common transaction types where the provider of the credit is queried as to certain parameters describing the agreement between the parties. The receiving portable appliance 2600 may iteratively or as a whole be queried as to the acceptance of the terms. VDE negotiation techniques described elsewhere in this application may be employed in a smart card transfer of electronic credit and/or currency to another VDE smart card or other VDE installation.

Such VDE electronic appliance 600/portable appliance 2600 credit transfer features would significantly reduce the overhead cost of managing certain electronic credit and/or currency activities by significantly automating these processes through extending the computerization of credit control and credit availability that was begun with credit cards and extended with debit cards. The automation of credit extension and/or currency transfer and the associated distributed processing advantages described, including the absence of any requirement for centralized processing and telecommunications during each transaction, truly make credit and/or currency, for many consumers and other electronic currency and/or credit users, an efficient, trusted, and portable commodity.

The portable appliance 2600 or other VDE electronic appliance 600, can, in one embodiment, also automate many tax collection functions. A VDE electronic appliance 600 may, with great security, record financial transactions, identify the nature of the transaction, and identify the required sales or related government transaction taxes, debit the taxes from the users available credit, and securely communicate this information to one or more government agencies directly at some interval (for example monthly), and/or securely transfer this information to, for example, a financial clearinghouse, which would then transfer one or more secure, encrypted (or unsecure, calculated by clearinghouse, or otherwise computed) information audit packets (e.g., VDE content containers and employing secure VDE communication techniques) to the one or more appropriate participating government agencies. The overall integrity and security of VDE 100 could ensure, in a coherent and centralized manner, that electronic reporting of tax related information (derived from one or more electronic commerce activities) would be valid and comprehensive. It could also act as a validating source of information on the transfer of sales tax collection (e.g., if, for example, said funds are transferred directly to the government by a commercial operation and/or transferred in a manner such that reported tax related information cannot be tampered with by other parties in a VDE pathway of tax

information handling). A government agency could select transactions randomly, or some subset or all of the reported transactions for a given commercial operation can be selected. This could be used to ensure that the commercial operation is actually paying to the government all appropriate collected funds required for taxes, and can also ensure that end-users are charged appropriate taxes for their transactions (including receipt of interest from bank accounts, investments, gifts, etc.

Portable appliance 2600 financial and tax processes could involve template mechanisms described elsewhere herein. While such an electronic credit and/or currency management capability would be particularly interesting if managed at least in part, through the use of a portable appliance 2600, credit and/or currency transfer and similar features would also be applicable for non-portable VDE electronic appliance 600's connected to or installed within a computer or other electronic device.

#### **User Notification Exception Interface ("Pop Up") 686**

As described above, the User Modification Exception Interface 686 may be a set of user interface programs for handling common VDE functions. These applications may be forms of VDE templates and are designed based upon certain assumptions regarding important options, specifically,

appropriate to a certain VDE user model and important  
messages that must be reported given certain events. A primary  
function of the "pop-up" user interface 686 is to provide a simple,  
consistent user interface to, for example, report metering events  
5 and exceptions (e.g., any condition for which automatic  
processing is either impossible or arguably undesirable) to the  
user, to enable the user to configure certain aspects of the  
operation of her electronic appliance 600 and, when appropriate,  
to allow the user to interactively control whether to proceed with  
10 certain transaction processes. If an object contains an exception  
handling method, that method will control how the "pop-up" user  
interface 686 handles specific classes of exceptions.

The "pop-user" interface 686 normally enables handling of  
15 tasks not dedicated to specific objects 300, such as for example:

- Logging onto an electronic appliance 600 and/or entering  
into a VDE related activity or class of activities,
- 20 • Configuring an electronic appliance 600 for a registered  
user, and/or generally for the installation, with regard to  
user preferences, and automatic handling of certain types  
of exceptions,

- Where appropriate, user selecting of meters for use with specific properties, and
- Providing an interface for communications with other electronic appliances 600, including requesting and/or for purchasing or leasing content from distributors, requesting clearinghouse credit and/or budgets from a clearinghouse, sending and/or receiving information to and/or from other electronic appliances, and so on.

10

Figure 72A shows an example of a common "logon" VDE electronic appliance 600 function that may use user interface 686. "Log-on" can be done by entering a user name, account name, and/or password. As shown in the provided example, a configuration option provided by the "pop-up" user interface 686 dialog can be "Login at Setup", which, if selected, will initiate a VDE Login procedure automatically every time the user's electronic appliance 600 is turned on or reset. Similarly, the "pop-up" user interface 686 could provide an interface option called "Login at Type" which, if selected, will initiate a procedure automatically every time, for example, a certain type of object or specific content type application is opened such as a file in a certain directory, a computer application or file with a certain identifying extension, or the like.

15

20

Figure 72B shows an example of a "pop-up" user interface  
686 dialog that is activated when an action by the user has been  
"trapped," in this case to warn the user about the amount of  
expense that will be incurred by the user's action, as well as to  
5 alert the user about the object 300 which has been requested and  
what that particular object will cost to use. In this example, the  
interface dialog provides a button allowing the user to request  
further detailed information about the object, including full text  
descriptions, a list of associated files, and perhaps a history of  
10 past usage of the object including any residual rights to use the  
object or associated discounts.

The "Cancel" button 2660 in Figure 72B cancels the user's  
trapped request. "Cancel" is the default in this example for this  
15 dialog and can be activated, for example, by the return and enter  
keys on the user's keyboard 612, by a "mouse click" on that  
button, by voice command, or other command mechanisms. The  
"Approve button" 2662, which must be explicitly selected by a  
mouse click or other command procedure, allows the user to  
20 approve the expense and proceed. The "More options" control  
2664 expands the dialog to another level of detail which provides  
further options, an example of which is shown in Figure 72C.



Figure 72C shows a secondary dialog that is presented to the user by the "pop-up" user interface 686 when the "More options" button 2664 in Figure 72B is selected by the user. As shown, this dialog includes numerous buttons for obtaining  
5 further information and performing various tasks.

In this particular example, the user is permitted to set "limits" such as, for example, the session dollar limit amount (field 2666), a total transaction dollar limit amount (field 2668), a  
10 time limit (in minutes) (field 2670), and a "unit limit" (in number of units such as paragraphs, pages, etc.) (field 2672). Once the user has made her selections, she may "click on" the OKAY button (2674) to confirm the limit selections and cause them to take effect.

15 Thus, pop-up user interface dialogues can be provided to specify user preferences, such as setting limits on budgets and/or other aspects of object content usage during any one session or over a certain duration of time or until a certain point in time.  
20 Dialogs can also be provided for selecting object related usage options such as selecting meters and budgets to be used with one or more objects. Selection of options may be applied to types (that is classes) of objects by associating the instruction with one or more identifying parameters related to the desired one or

more types. User specified configuration information can set default values to be used in various situations, and can be used to limit the number or type of occasions on which the user's use of an object is interrupted by a "pop-up" interface 686 dialog. For example, the user might specify that a user request for VDE protected content should be automatically processed without interruption (resulting from an exceptions action) if the requested processing of information will not cost more than \$25.00 and if the total charge for the entire current session (and/or day and/or week, etc.) is not greater than \$200.00 and if the total outstanding and unpaid charge for use hasn't exceeded \$2500.00.

Pop-up user interface dialogs may also be used to notify the user about significant conditions and events. For example, interface 686 may be used to:

- remind the user to send audit information to a clearinghouse,
- inform a user that a budget value is low and needs replenishing,
- remind the user to back up secure database 610, and

- inform the user about expirations of PERCs or other dates/times events.

Other important "pop-up" user interface 686 functions  
5 include dialogs which enable flexible browsing through libraries  
of properties or objects available for licensing or purchase, either  
from locally stored VDE protected objects and/or from one or  
more various, remotely located content providers. Such function  
may be provided either while the user's computer is connected to  
10 a remote distributor's or clearinghouse's electronic appliance 600,  
or by activating an electronic connection to a remote source after  
a choice (such as a property, a resource location, or a class of  
objects or resources is selected). A browsing interface can allow  
this electronic connection to be made automatically upon a user  
15 selection of an item, or the connection itself can be explicitly  
activated by the user. See Figure 72D for an example of such a  
"browsing" dialog.

#### Smart Objects

20 VDE 100 extends its control capabilities and features to  
"intelligent agents." Generally, an "intelligent agent" can act as  
an emissary to allow a process that dispatches it to achieve a  
result the originating process specifies. Intelligent agents that  
are capable of acting in the absence of their dispatch process are

particularly useful to allow the dispatching process to access, through its agent, the resources of a remote electronic appliance. In such a scenario, the dispatch process may create an agent (e.g., a computer program and/or control information associated with a computer program) specifying a particular desired task(s), and dispatch the agent to the remote system. Upon reaching the remote system, the "agent" may perform its assigned task(s) using the remote system's resources. This allows the dispatch process to, in effect, extend its capabilities to remote systems where it is not present.

Using an "agent" in this manner increases flexibility. The dispatching process can specify, through its agent, a particular desired task(s) that may not exist or be available on the remote system. Using such an agent also provides added trustedness; the dispatch process may only need to "trust" its agent, not the entire remote system. Agents have additional advantages.

Software agents require a high level of control and accountability to be effective, safe and useful. Agents in the form of computer viruses have had devastating effects worldwide. Therefore, a system that allows an agent to access it should be able to control it or otherwise prevent the agent from damaging important resources. In addition, systems allowing themselves

to be accessed by an agent should sufficiently trust the agent  
and/or provide mechanisms capable of holding the true  
dispatcher of the agent responsible for the agent's activities.  
Similarly, the dispatching process should be able to adequately  
5 limit and/or control the authority of the agents it dispatches or  
else it might become responsible for unforeseen activities by the  
agent (e.g., the agent might run up a huge bill in the course of  
following imprecise instructions it was given by the process that  
dispatched it).

10

These significant problems in using software agents have  
not be adequately addressed in the past. The open, flexible  
control structures provided by VDE 100 addresses these  
problems by providing the desired control and accountability for  
15 software agents (e.g., agent objects). For example, VDE 100  
positively controls content access and usage, provides guarantee  
of payment for content used, and enforces budget limits for  
accessed content. These control capabilities are well suited to  
controlling the activities of a dispatched agent by both the  
20 process that dispatches the agent and the resource accessed by  
the dispatched agent.

One aspect of the preferred embodiment provided by the  
present invention provides a "smart object" containing an agent.

Generally, a "smart object" may be a VDE object 300 that contains some type(s) of software programs ("agents") for use with VDE control information at a VDE electronic appliance 600.

A basic "smart object" may comprise a VDE object 300 that, for example, contains (physically and/or virtually):

a software agent, and

at least one rule and/or control associated with the software agent that governs the agent's operation.

Although this basic structure is sufficient to define a "smart object," Figure 73 shows a combination of containers and control information that provides one example of a particularly advantageous smart object structure for securely managing and controlling the operation of software agents.

As shown in Figure 73, a smart object 3000 may be constructed of a container 300, within which is embedded one or more further containers (300z, 300y, etc.). Container 300 may further contain rules and control information for accessing and using these embedded containers 300z, 300y, etc. Container 300z embedded in container 300 is what makes the object 3000 a "smart object." It contains an "agent" that is managed and controlled by VDE 100.

5       The rules and control information 806f associated with  
container 300z govern the circumstances under which the agent  
may be released and executed at a remote VDE site, including  
any limitations on execution based on the cost of execution for  
example. This rule and control information may be specified  
entirely in container 300z, and/or may be delivered as part of  
container 300, as part of another container (either within  
container 300 or a separately deliverable container), and/or may  
be already present at the remote VDE site.

10

      The second container 300y is optional, and contains  
content that describes the locations at which the agent stored in  
container 300z may be executed. Container 300y may also  
contain rules and control information 806e that describe the  
15       manner in which the contents of container 300y may be used or  
altered. This rule and control information 806e and/or further  
rules 300y(1) also contained within container 300y may describe  
searching and routing mechanisms that may be used to direct  
the smart object 3000 to a desired remote information resource.  
20       Container 300y may contain and/or reference rules and control  
information 300y(1) that specify the manner in which searching  
and routing information use and any changes may be paid for.

Container 300x is an optional content container that is initially "empty" when the smart object 3000 is dispatched to a remote site. It contains rules and control information 300x(1) for storing the content that is retrieved by the execution of the agent  
5 contained in container 300z. Container 300x may also contain limits on the value of content that is stored in the retrieval container so as to limit the amount of content that is retrieved.

Other containers in the container 300 may include  
10 administrative objects that contain audit and billing trails that describe the actions of the agent in container 300z and any charges incurred for executing an agent at a remote VDE node. The exact structure of smart object 3000 is dependent upon the type of agent that is being controlled, the resources it will need  
15 for execution, and the types of information being retrieved.

The smart object 3000 in the example shown in Figure 73 may be used to control and manage the operation of an agent in VDE 100. The following detailed explanation of an example  
20 smart object transaction shown in Figure 74 may provide a helpful, but non-limiting illustration. In this particular example, assume a user is going to create a smart object 3000 that performs a library search using the "Very Fast and Efficient" software agent to search for books written about some subject of



interest (e.g., "fire flies"). The search engine is designed to return a list of books to the user. The search engine in this example may spend no more than \$10.00 to find the appropriate books, may spend no more than \$3.00 in library access or communications charges to get to the library, and may retrieve no more than \$15.00 in information. All information relating to the search or use is to be returned to the user and the user will permit no information pertaining to the user or the agent to be released to a third party.

In this example, a dispatching VDE electronic appliance 3010 constructs a smart object 3000 like the one shown in Figure 73. The rule set in 806a is specified as a control set that contains the following elements:

1. a smart\_agent\_execution event that specifies the smart agent is stored in embedded container 300z and has rules controlling its execution specified in that container;
2. a smart\_agent\_use event that specifies the smart agent will operate using information and parameters stored in container 300;

3. a routing\_use event that specifies the information routing information is stored in container 300y and has rules controlling this information stored in that container;

5

4. an information\_write event that specifies information written will be stored in container 300y, 300x, or 300w depending on its type (routing, retrieved, or administrative), and that these containers have independent rules that control how information is written into them.

10

The rule set in control set 806b contains rules that specify the rights desired by this smart object 3000. Specifically, this control set specifies that the software agent desires:

15

1. A right to use the "agent execution" service on the remote VDE site. Specific billing and charge information for this right is carried in container 300z.

20

2. A right to use the "software description list" service on the remote VDE site. Specific billing and charge

information for this for this right is carried in  
container 300y.

- 5
3. A right to use an "information locator service" on a remote VDE site.
  4. A right to have information returned to the user without charge (charges to be incurred on release of information and payment will be by a VISA budget)
  - 10 5. A right to have all audit information returned such that it is readable only by the sender.

15 The rule set in control set 806c specifies that container 300w specifies the handling of all events related to its use. The rule set in control set 806d specifies that container 300x specifies the handling of all events related to its use. The rule set in control set 806e specifies that container 300y specifies the handling of all events related to its use. The rule set in control  
20 set 806f specifies that container 300z specifies the handling of all events related to its use.

Container 300z is specified as containing the "Very Fast and Efficient" agent content, which is associated with the following rules set:

- 5                   1.    A use event that specifies a meter and VISA budget that limits the execution to \$10.00 charged against the owner's VISA card. Audits of usage are required and will be stored in object 300w under control information specified in that object.

10

After container 300z and its set are specified, they are constructed and embedded in the smart object container 300.

15                   Container 300y is specified as a content object with two types of content. Content type A is routing information and is read/write in nature. Content type A is associated with a rules set that specifies:

- 20                   1.    A use event that specifies no operation for the release of the content. This has the effect of not charging for the use of the content.
2.    A write event that specifies a meter and a VISA budget that limits the value of writing to \$3.00. The

billing method used by the write is left unspecified and will be specified by the control method that uses this rule.

- 5                    3.     Audits of usage are required and will be stored in object 300w under control information specified in that object.

10                   Content type B is information that is used by the software agent to specify parameters for the agent. This content is specified as the string "fire fly" or "fire flies". Content type B is associated with the following rule set:

- 15                   1.     A use event that specifies that the use may only be by the software agent or a routing agent. The software agent has read only permission, the routing agent has read/write access to the information.
- 20                   There are no charges associated with using the information, but two meters; one by read and one by write are kept to track use of the information by various steps in the process.

2. Audits of usage are required and will be stored in object 300w under control information specified in that object.

5           After container 300y and its control sets are specified, they are constructed and embedded in the smart object container 300.

          Container 300x is specified as a content object that is empty of content. It contains a control set that contains the  
10       following rules:

1. A write\_without\_billing event that specifies a meter and a general budget that limits the value of writing to \$15.00.

15

2. Audits of usage are required and will be stored in object 300w under control information specified in that object.

20

3. An empty use control set that may be filled in by the owner of the information using predefined methods (method options).

After container 300x and its control sets are specified, they are constructed and embedded in the smart object container 300.

5 Container 300w is specified as an empty administrative object with a control set that contains the following rules:

1. A use event that specifies that the information contained in the administrative object may only be released to the creator of smart object container 300.
- 10 2. No other rules may be attached to the administrative content in container 300w.

After container 300w and its control sets are specified, they are constructed and embedded in the smart object container 15 300.

At this point, the smart object has been constructed and is ready to be dispatched to a remote VDE site. The smart object is sent to a remote VDE site (e.g., using electronic mail or another transport mechanism) that contains an information locator 20 service 3012 via path 3014. The smart object is registered at the remote site 3012 for the "item locator service." The control set in container related to "item locator service" is selected and the rules contained within it activated at the remote site 3012. The

remote site 3012 then reads the contents of container 300y under the control of rule set 806f and 300y(1), and permits writes of a list of location information into container 300y pursuant to these rules. The item locator service writes a list of three items into  
5 the smart object, and then "deregisters" the smart object (now containing the location information) and sends it to a site 3016 specified in the list written to the smart object via path 3018. In this example, the user may have specified electronic mail for transport and a list of remote sites that may have the desired  
10 information is stored as a forwarding list.

The smart object 3000, upon arriving at the second remote site 3016, is registered with that second site. The site 3016 provides agent execution and software description list services  
15 compatible with VDE as a service to smart objects. It publishes these services and specifies that it requires \$10.00 to start the agent and \$20/piece for all information returned. The registration process compares the published service information against the rules stored within the object and determines that an  
20 acceptable overlap does not exist. Audit information for all these activities is written to the administrative object 300w. The registration process then fails (the object is not registered), and the smart object is forwarded by site 3016 to the next VDE site 3020 in the list via path 3022.



The smart object 3000, upon arriving at the third remote site 3020, is registered with that site. The site 3020 provides agent execution and software description list services compatible with VDE as a service to smart objects. It publishes these services and specifies that it requires \$1.00 to start the agent and \$0.50/piece for all information returned. The registration process compares the published service information against the rules stored within the object and determines that an acceptable overlap exists. The registration process creates a URT that specifies the agreed upon control information. This URT is used in conjunction with the other control information to execute the software agent under VDE control.

The agent software starts and reads its parameters out of container 300y. It then starts searching the database and obtains 253 "hits" in the database. The list of hits is written to container 300x along with a completed control set that specifies the granularity of each item and that each item costs \$0.50. Upon completion of the search, the budget for use of the service is incremented by \$1.00 to reflect the use charge for the service. Audit information for all these activities is written to the administrative object 300w.

The remote site 3020 returns the now "full" smart object 3000 back to the original sender (the user) at their VDE node 3010 via path 3024. Upon arrival, the smart object 3000 is registered and the database records are available. The control information specified in container 300x is now a mix of the original control information and the control information specified by the service regarding remote release of their information. The user then extracts 20 records from the smart object 3000 and has \$10.00 charged to her VISA budget at the time of extraction.

In the above smart agent VDE examples, a certain organization of smart object 3000 and its constituent containers is described. Other organizations of VDE and smart object related control information and parameter data may be created and may be used for the same purposes as those ascribed to object 3000 in the above example.

### **Negotiation and Electronic Contracts**

An electronic contract is an electronic form of an agreement including rights, restrictions, and obligations of the parties to the agreement. In many cases, electronic agreements may surround the use of digitally provided content; for example, a license to view a digitally distributed movie. It is not required, however, that an electronic agreement be conditioned on the

presence or use of electronic content by one or more parties to the agreement. In its simplest form, an electronic agreement contains a right and a control that governs how that right is used.

5

Electronic agreements, like traditional agreements, may be negotiated between their parties (terms and conditions submitted by one or more parties may simply be accepted (cohesion contract) by one or more other parties and/or such other parties may have the right to select certain of such terms and conditions (while others may be required)). Negotiation is defined in the dictionary as "the act of bringing together by mutual agreement." The preferred embodiment provides electronic negotiation processes by which one or more rights and associated controls can be established through electronic automated negotiation of terms. Negotiations normally require a precise specification of rights and controls associated with those rights. PERC and URT structures provide a mechanism that may be used to provide precise electronic representations of rights and the controls associated with those rights. VDE thus provides a "vocabulary" and mechanism by which users and creators may specify their desires. Automated processes may interpret these desires and negotiate to reach a common middle ground based on these desires. The results of said negotiation

10

15

20

may be concisely described in a structure that may be used to control and enforce the results of the electronic agreement. VDE further enables this process by providing a secure execution space in which the negotiation process(es) are assured of integrity and confidentiality in their operation. The negotiation process(es) may also be executed in such a manner that inhibits external tampering with the negotiation.

A final desirable feature of agreements in general (and electronic representations of agreements in particular) is that they be accurately recorded in a non-repudiatable form. In traditional terms, this involves creating a paper document (a contract) that describes the rights, restrictions, and obligations of all parties involved. This document is read and then signed by all parties as being an accurate representation of the agreement. Electronic agreements, by their nature, may not be initially rendered in paper. VDE enables such agreements to be accurately electronically described and then electronically signed to prevent repudiation. In addition, the preferred embodiment provides a mechanism by which human-readable descriptions of terms of the electronic contract can be provided.

VDE provides a concise mechanism for specifying control sets that are VDE site interpretable. Machine interpretable

mechanisms are often not human readable. VDE often operates the negotiation process on behalf of at least one human user. It is thus desirable that the negotiation be expressible in "human readable form." VDE data structures for objects, methods, and load modules all have provisions to specify one or more DTDs within their structures. These DTDs may be stored as part of the item or they may be stored independently. The DTD describes one or more data elements (MDE, UDE, or other related data elements) that may contain a natural language description of the function of that item. These natural language descriptions provide a language independent, human readable description for each item. Collections of items (for example, a BUDGET method) can be associated with natural language text that describes its function and forms a term of an electronically specified and enforceable contract. Collections of terms (a control set) define a contract associated with a specific right. VDE thus permits the electronic specification, negotiation, and enforcement of electronic contracts that humans can understand and adhere to.

20

VDE 100 enables the negotiation and enforcement of electronic contracts in several ways:

- it enables a concise specification of rights and control information that permit a common vocabulary and procedure for negotiation,
- 5 ● it provides a secure processing environment within which to negotiate,
- it provides a distributed environment within which rights and control specifications may be securely distributed,
- 10 ● it provides a secure processing environment in which negotiated contracts may be electronically rendered and signed by the processes that negotiate them, and
- 15 ● it provides a mechanism that securely enforces a negotiated electronic contract.

## 20 **Types of Negotiations**

A simple form of a negotiation is a demand by one party to form an "adhesion" contract. There are few, if any, options that may be chosen by the other party in the negotiation. The recipient of the demand has a simple option; she may accept or

reject the terms and conditions (control information) in the demand. If she accepts the conditions, she is granted rights subject to the specified control information. If she rejects the conditions, she is not granted the rights. PERC and URT  
5 structures may support negotiation by demand; a PERC or control set from a PERC may be presented as a demand, and the recipient may accept or reject the demand (selecting any permitted method options if they are presented).

10 A common example of this type of negotiation today is the purchase of software under the terms of a "shrink-wrap license." Many widely publicized electronic distribution schemes use this type of negotiation. CompuServe is an example of an on-line service that operates in the same manner. The choice is simple:  
15 either pay the specified charge or don't use the service or software. VDE supports this type of negotiation with its capability to provide PERCs and URTs that describe rights and control information, and by permitting a content owner to provide a REGISTER method that allows a user to select from a  
20 set of predefined method options. In this scenario, the REGISTER method may contain a component that is a simplified negotiation process.

A more complex form of a negotiation is analogous to "haggling." In this scenario, most of the terms and conditions are fixed, but one or more terms (e.g., price or payment terms) are not. For these terms, there are options, limits, and elements that may be negotiated over. A VDE electronic negotiation between two parties may be used to resolve the desired, permitted, and optional terms. The result of the electronic negotiation may be a finalized set of rules and control information that specify a completed electronic contract. A simple example is the scenario for purchasing software described above adding the ability of the purchaser to select a method of payment (VISA, Mastercard, or American Express). A more complex example is a scenario for purchasing information in which the price paid depends on the amount of information about the user that is returned along with a usage audit trail. In this second example, the right to use the content may be associated with two control sets. One control set may describe a fixed ("higher") price for using the content. Another control set may describe a fixed ("lower") price for using the content with additional control information and field specifications requiring collection and return the user's personal information. In both of these cases, the optional and permitted fields and control sets in a PERC may describe the options that may be selected as part of the negotiation. To perform the negotiation, one party may propose a control set containing



specific fields, control information, and limits as specified by a PERC; the other party may pick and accept from the control sets proposed, reject them, or propose alternate control sets that might be used. The negotiation process may use the permitted, required, and optional designations in the PERC to determine an acceptable range of parameters for the final rule set. Once an agreement is reached, the negotiation process may create a new PERC and/or URT that describes the result of the negotiation. The resulting PERCs and/or URTs may be "signed" (e.g., using digital signatures) by all of the negotiation processes involved in the negotiation to prevent repudiation of the agreement at a later date.

Additional examples of negotiated elements are: electronic cash, purchase orders, purchase certificates (gift certificates, coupons), bidding and specifications, budget "rollbacks" and reconciliation, currency exchange rates, stock purchasing, and billing rates.

A set of PERCs that might be used to support the second example described above is presented in Figures 75A (PERC sent by the content owner), 75B (PERC created by user to represent their selections and rights), and 75C (PERC for controlling the negotiation process). These PERCs might be used in conjunction

with any of the negotiation process(es) and protocols described later in this section.

Figure 75A shows an example of a PERC 3100 that might be created by a content provider to describe their rights options. In this example, the PERC contains information regarding a single USE right. Two alternate control sets 3102a, 3102b are presented for this right in the example. Control set 3102a permits the use of the content without passing back information about the user, and another control set 3102b permits the use of the content and collects "response card" type information from the user. Both control sets 3102a, 3102b may use a common set of methods for most of the control information. This common control information is represented by a CSR 3104 and CS0 3106.

Control set 3102a in this PERC 3100 describes a mechanism by which the user may obtain the content without providing any information about its user to the content provider. This control set 3102a specifies a well-known vending control method and set of required methods and method options. Specifically, in this example, control set 3102a defines a BUDGET method 3108 (e.g., one of VISA, Mastercard, or American Express) and it defines a BILLING method 3110 that specifies a charge (e.g., a one-time charge of \$100.00).

Control set 3102b in this PERC 3100 describes another mechanism by which the user may obtain the content. In this example, the control set 3102b specifies a different vending control method and a set of required methods and method options. This second control set 3102b specifies a BUDGET method 3112 (e.g., one of VISA, Mastercard, or American Express), a BILLING method 3116 that specifies a charge (e.g., a lesser one-time charge such as \$25.00) and an AUDIT method 3114 that specifies a set of desired and required fields. The required and desired field specification 3116 may take the form of a DTD specification, in which, for example, the field names are listed.

The content creator may "prefer" one of the two control sets (e.g., control set 2) over the other one. If so, the "preferred" control set may be "offered" first in the negotiation process, and withdrawn in favor of the "non-preferred" control set if the other party to the negotiation "rejects" the "preferred" control set.

In this example, these two control sets 3102a, 3102b may share a common BUDGET method specification. The BUDGET method specification may be included in the CSR 3104 or CS0 3106 control sets if desired. Selecting control set 3102a (use with no information passback) causes a unique component assembly

to be assembled as specified by the PERC 3100. Specifically, in this example it selects the "Vending" CONTROL method 3118, the BILLING method 3110 for a \$100 fixed charge, and the rest of the control information specified by CSR 3104 and CS0 3106.

5 It also requires the user to specify her choice of acceptable BUDGET method (e.g., from the list including VISA, Mastercard, and American Express). Selecting control set 3102b assembles a different component assembly using the "Vending with 'response card'" CONTROL method 3120, the BILLING method 3116 (e.g.,

10 for a \$25 fixed charge), an AUDIT method 3114 that requires the fields listed in the Required Fields DTD 3116. The process may also select as many of the fields listed in the Desired Fields DTD 3116 as are made available to it. The rest of the control information is specified by CSR 3104 and CS0 3106. The

15 selection of control set 3102b also forces the user to specify their choice of acceptable BUDGET methods (e.g., from the list including VISA, Mastercard, and American Express).

Figure 75B shows an example of a control set 3125 that

20 might be used by a user to specify her desires and requirements in a negotiation process. This control set has a USE rights section 3127 that contains an aggregated CSR budget specification 3129 and two optional control sets 3131a, 3131b for use of the content. Control set 3131a requires the use of a

specific CONTROL method 3133 and AUDIT method 3135. The specified AUDIT method 3135 is parameterized with a list of fields 3137 that may be released in the audit trail. Control set 3131a also specifies a BILLING method 3139 that can cost no more than a certain amount (e.g., \$30.00). Control set 3131b in this example describes a specific CONTROL method 3141 and may reference a BILLING method 3143 that can cost no more than a certain amount (e.g., \$150.00) if this option is selected.

Figure 75E shows a more high-level view of an electronic contract 3200 formed as a "result" of a negotiation process as described above. Electronic contract 3200 may include multiple clauses 3202 and multiple digital signatures 3204. Each clause 3202 may comprise a PERC/URT such as item 3160 described above and shown in Figure 75D. Each "clause" 3202 of electronic contract 3200 thus corresponds to a component assembly 690 that may be assembled and executed by a VDE electronic appliance 600. Just as in normal contracts, there may be as many contract clauses 3202 within electronic contract 3200 as is necessary to embody the "agreement" between the "parties." Each of clauses 3202 may have been electronically negotiated and may thus embody a part of the "agreement" (e.g., a "compromise") between the parties. Electronic contract 3200 is "self-executing" in the sense that it may be literally executed by

a machine, i.e., a VDE electronic appliance 600 that assembles component assemblies 690 as specified by various electronic clauses 3202. Electronic contract 3200 may be automatically "enforced" using the same VDE mechanisms discussed above

5 that are used in conjunction with any component assembly 690. For example, assuming that a clause 3202(2) corresponds to a payment or BILLING condition or term, its corresponding component assembly 690 when assembled by a user's VDE electronic appliance 600 may automatically determine whether

10 conditions are right for payment and, when they are, automatically access an appropriate payment mechanism (e.g., a virtual "credit card" object for the user) to arrange that payment to be made. As another example, assuming that electronic contract clause N 3202(N) corresponds to a user's obligation to

15 provide auditing information to a particular VDE participant, electronic contract 3200 will cause VDE electronic appliance 600 to assemble a corresponding component assembly 690 that may, for example, access the appropriate audit trails within secure database 610 and provide them in an administrative object to the

20 correct participant. Figure 75F shows that clause 3202(N) may, for example, specify a component assembly 690 that arranges for multiple steps in a transaction 3206 to occur. Some of these steps (e.g., step 3208(4), 3208(5)) may be conditional on a test (e.g., 3208(3)) such as, for example, whether content usage has

exceeded a certain amount, whether a certain time period has expired, whether a certain calendar date has been reached, etc.

5 Digital signatures 3204 shown in the Figure 75E electronic contract 3200 may comprise, for example, conventional digital signatures using public key techniques as described above. Some electronic contracts 3200 may not bear any digital signatures 3204. However, it may be desirable to require the electronic appliance 600 of the user who is a party to the electronic contract 10 3200 to digitally "sign" the electronic contract so that the user cannot later repudiate the contract, for evidentiary purposes, etc. Multiple parties to the same contract may each digitally "sign" the same electronic contract 3200 similarly to the way multiple parties to a contract memorialized in a written instrument use 15 an ink pen to sign the instrument.

Although each of the clauses 3202 of electronic contract 3200 may ultimately correspond to a collection of data and code that may be executed by a PPE 650, there may in some instances 20 be a need for rendering a human readable version of the electronic contract. This need can be accommodated by, as mentioned above, providing text within one or more DTDs associated with the component assembly or assemblies 690 used to "self-execute" the contract. Such text might, for example,

describe from a functional point of view what the corresponding electronic contract clause 3202 means or involves, and/or might describe in legally enforceable terms what the legal obligation under the contract is or represents. "Templates" (described elsewhere herein) might be used to supply such text from a text library. An expert system and/or artificial intelligence capability might be used to impose syntax rules that bind different textual elements together into a coherent, humanly readable contract document. Such text could, if necessary, be reviewed and modified by a "human" attorney in order to customize it for the particular agreement between the parties and/or to add further legal obligations augmenting the "self-executing" electronic obligations embodied within and enforced by the associated component assemblies 690 executing on a VDE electronic appliance 600. Such text could be displayed automatically or on demand upon execution of the electronic contract, or it could be used to generate a printed humanly-readable version of the contract at any time. Such a document version of the electronic contract 3200 would not need to be signed in ink by the parties to the agreement (unless desired) in view of the fact that the digital signatures 3204 would provide a sufficiently secure and trusted evidentiary basis for proving the parties' mutual assent to all the terms and conditions within the contract.



In the preferred embodiment, the negotiation process executes within a PPE 650 under the direction of a further PERC that specifies the process. Figure 75C shows an example of a PERC 3150 that specifies a negotiation process. The PERC 3150  
5 contains a single right 3152 for negotiation, with two permitted control sets 3154a, 3154b described for that right. The first control set 3154a may be used for a "trusted negotiation"; it references the desired negotiation CONTROL method ("Negotiate") 3156 and references (in fields 3157a, 3157b) two  
10 UDEs that this CONTROL method will use. These UDEs may be, for example, the PERCs 3100, 3125 shown in Figures 75A and 75B. The second control set 3154b may be used by "multiple negotiation" processes to manage the negotiation, and may provide two negotiation methods: "Negotiate1," and  
15 "Negotiate2". Both negotiation processes may be described as required methods ("Negotiate1" and "Negotiate2") 3156, 3158 that take respective PERCs 3100, 3125 as their inputs. The CONTROL method 3158 for this control set in this example may specify the name of a service that the two negotiation processes  
20 will use to communicate with each other, and may also manage the creation of the URT resulting from the negotiation.

When executed, the negotiation process(es) specified by the PERC 3150 shown in Figure 75C may be provided with the

PERCs 3100, 3125 as input that will be used as the basis for negotiation. In this example, the choice of negotiation process type (trusted or multiple) may be made by the executing VDE node. The PERC 3150 shown in Figure 75C might be, for  
5 example, created by a REGISTER method in response to a register request from a user. The process specified by this PERC 3150 may then be used by a REGISTER method to initiate negotiation of the terms of an electronic contract.

10           During this example negotiation process, the PERCs 3100, 3125 shown in Figures 75A and 75B act as input data structures that are compared by a component assembly created based on PERC 3150 shown in Figure 35C. The component assembly specified by the control sets may be assembled and compared,  
15 starting with required "terms," and progressing to preferred/desired "terms" and then moving on to permitted "terms," as the negotiation continues. Method option selections are made using the desired method and method options specified in the PERCs 3100, 3125. In this example, a control set for the  
20 PERC 3100 shown in Figure 75A may be compared against the PERC 3125 shown in Figure 75B. If there is a "match," the negotiation is successfully concluded and "results" are generated.

In this embodiment, the results of such negotiation will generally be written as a URT and "signed" by the negotiation process(es) to indicate that an agreement has been reached. These electronic signatures provide the means to show that a (virtual) "meeting of minds" was reached (one of the traditional legal preconditions for a contract to exist). An example of the URT 3160 that would have been created by the above example is shown in Figure 75D.

This URT 3160 (which may itself be a PERC 808) includes a control set 3162 that reflects the "terms" that were "agreed upon" in the negotiation. In this example, the "agreed upon" terms must "match" terms required by input PERCs 3100, 3125 in the sense that they must be "as favorable as" the terms required by those PERCs. The negotiation result shown includes, for example, a "negotiated" control set 3162 that in some sense corresponds to the control set 3102a of the Figure 75A PERC 3100 and to the control set 3131a of the Figure 75B control set 3125. Resulting "negotiated" control set 3162 thus includes a required BUDGET method 3164 that corresponds to the control set 3125 desired BUDGET method 3142 but which is "within" the range of control sets allowed by control set 3100 required BUDGET method 3112. Similarly, resulting negotiated control set 3162 includes a required AUDIT method 3166 that

complies with the requirements of both PERC 3100 required  
 AUDIT method 3114 and PERC 3125 required AUDIT method  
 3135. Similarly, resulting negotiated control set 3162 includes a  
 required BILLING method 3170 that "matches" or complies with  
 5 each of PERC 3100 required BILLING method 3116 and PERC  
 3125 required BILLING method 3170.

Another class of negotiation is one under which no rules  
 are fixed and only the desired goals are specified. The  
 10 negotiation processes for this type of negotiation may be very  
 complex. It may utilize artificial intelligence, fuzzy logic, and/or  
 related algorithms to reach their goals. VDE supports these  
 types of processes by providing a mechanism for concisely  
 specifying rights, control information, fields and goals (in the  
 15 form of desired rights, control information, and fields). Goals for  
 these types of processes might be specified as one more control  
 sets that contain specific elements that are tagged as optional,  
 permitted, or desired.

## 20 **Types of Negotiations**

Negotiations in the preferred embodiment may be  
 structured in any of the following ways:

1. shared knowledge
2. trusted negotiator

### 3. "zero-based" knowledge

"Shared knowledge" negotiations are based on all parties knowing all of the rules and constraints associated with the negotiation. Demand negotiations are a simple case of shared knowledge negotiations; the demander presents a list of demands that must be accepted or rejected together. The list of demands comprises a complete set of knowledge required to accept or reject each item on the list. VDE enables this class of negotiation to occur electronically by providing a mechanism by which demands may be encoded, securely passed, and securely processed between and with secure VDE subsystems using VDE secure processing, and communication capabilities. Other types of shared knowledge negotiations employed by VDE involve the exchange of information between two or more negotiating parties; the negotiation process(es) can independently determine desired final outcome(s) based on their independent priorities. The processes can then negotiate over any differences. Shared knowledge negotiations may require a single negotiation process (as in a demand type negotiation) or may involve two or more cooperative processes. Figures 76A and 76B illustrate scenarios in which one and two negotiation processes are used in a shared knowledge negotiation.

Figure 76A shows a single negotiation process 3172 that takes any number of PERCs 808 (e.g., supplied by different parties) as inputs to the negotiation. The negotiation process 3172 executes at a VDE node under supervision of "Negotiation Process Rules and Control information" that may be supplied by a further PERC (e.g., PERC 3150 shown in Figure 75C). The process 3172 generates one or more PERCs/URTs 3160 as results of the negotiation.

Figure 76B shows multiple negotiation processes 3172A-3172N each of which takes as input a PERC 808 from a party and a further PERC 3150 that controls the negotiation process, and each of which generates a negotiated "result" PERC/URT 3160 as output. Processes 3172A-3172N may execute at the same or different VDE nodes and may communicate using a "negotiation protocol."

Single and multiple negotiation processes may be used for specific VDE sites. The negotiation processes are named, and can be accessed using well known method names. PERCs and URTs may be transported in administrative or smart objects to remote VDE sites for processing at that site, as may the control PERCs and REGISTER method that controls the negotiation.

Multiple negotiation processes require the ability to communicate between these processes 3172; including secure communication between secure processes that are present at physically separate VDE sites (secure subsystems). VDE  
5 generalizes the inter-process communication into a securely provided service that can be used if the configuration requires it. The inter-process communication uses a negotiation protocol to exchange information about rule sets between processes 3172. An example of a negotiation protocol includes the following  
10 negotiation "primitives":

	WANT	Want a set of terms and conditions
	ACCEPT	Accept a set of terms and conditions
15	REJECT	Reject a set of terms and conditions
	OFFER	Offer a set of terms and conditions in exchange for other terms and conditions
20	HAVE	Assert a set of terms and conditions are possible or desirable
	QUIT	Assert the end of the negotiation without reaching an agreement

AGREEMENT      Conclude the negotiation and pass the  
rule set for signature

5      The WANT primitive takes rights and control set (or parts  
of control sets) information, and asserts to the other process(es)  
3172 that the specified terms are desired or required. Demand  
negotiations are a simple case of a WANT primitive being used to  
assert the demand. This example of a protocol may introduce a  
refined form of the WANT primitive, REQUIRE. In this  
10      example, REQUIRE allows a party to set terms that she decides  
are necessary for a contract to be formed, WANT may allow the  
party to set terms that are desirable but not essential. This  
permits a distinction between "must have" and "would like to  
have."

15

In this example, WANT primitives must always be  
answered by an ACCEPT, REJECT, or OFFER primitive. The  
ACCEPT primitive permits a negotiation process 3172 to accept  
a set of terms and conditions. The REJECT primitive permits a  
20      process 3172 to reject an offered set of terms and conditions.  
Rejecting a set of required terms and conditions may terminate  
the negotiation. OFFER permits a counter-offer to be made.



The HAVE, QUIT, and AGREEMENT primitives permit the negotiation protocols to pass information about rule sets. Shared knowledge negotiations may, for example, start with all negotiation processes 3172A-3172N asserting HAVE (my PERC) to the other processes. HAVE is also used when an impasse is reached and one process 3172 needs to let the other process 3172 know about permitted options. QUIT signals an unsuccessful end of the negotiation without reaching an agreement, while AGREEMENT signals a successful end of an agreement and passes the resulting "negotiated" PERC/URT 3160 to the other process(es) 3172 for signature.

In "trusted negotiator" negotiations, all parties provide their demands and preferences to a "trusted" negotiator and agree to be bound by her decision. This is similar to binding arbitration in today's society. VDE enables this mode of negotiation by providing an environment in which a "trusted" negotiation service may be created. VDE provides not only the mechanism by which demands, desires, and limits may be concisely specified (e.g., in PERCs), but in which the PERCs may be securely transferred to a "trusted" negotiation service along with a rule set that specifies how the negotiation will be conducted, and by providing a secure execution environment so that the negotiation process may not be tampered with. Trusted

negotiator services can be used at VDE sites where the integrity of the site is well known. Remote trusted negotiation services can be used by VDE sites that do not possess sufficient computing resources to execute one or more negotiation process(es); they can establish a communication link to a VDE site that provides this service and permits the service to handle the negotiation on their behalf.

"Zero-based" knowledge negotiations share some characteristics of the zero-based knowledge protocols used for authentication. It is well understood in the art how to construct a protocol that can determine if a remote site is the holder of a specific item without exchanging or exposing the item. This type of protocol can be constructed between two negotiation processes operating on at least one VDE site using a control set as their knowledge base. The negotiation processes may exchange information about their control sets, and may make demands and counter proposals regarding using their individual rule sets. For example, negotiation process A may communicate with negotiation process B to negotiate rights to read a book. Negotiation process A specifies that it will pay not more than \$10.00 for rights to read the book, and prefers to pay between \$5.00 and \$6.00 for this right. Process A's rule set also specifies that for the \$5.00 option, it will permit the release of the reader's

name and address. Process B's rule set specifies that it wants \$50.00 for rights to read the book, and will provide the book for \$5.50 if the user agrees to release information about himself.

The negotiation might go something like this:

5	<div style="display: flex; justify-content: space-between;"> <span>Process A</span> <span>&lt;--- &gt; Process B</span> </div>
10	<div style="display: flex; justify-content: space-between;"> <span>Want (right to read, unrestricted)</span> <span>----&gt;</span> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <span></span> <span>&lt;---- Have(right to read, unrestricted, \$50)</span> </div>
15	<div style="display: flex; justify-content: space-between;"> <span>Offer (right to read, tender user info)</span> <span>----&gt;</span> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <span></span> <span>&lt; ---- Have(right to read, tender user info, \$5.50)</span> </div>
20	<div style="display: flex; justify-content: space-between;"> <span>Accept(right to read, tender user info, \$5.50)</span> <span>----- &gt;</span> </div>

In the above example, process A first specifies that it desires the right to read the book without restrictions or other information release. This starting position is specified as a rights option in the PERC that process A is using as a rule. Process B checks its rules and determines that an unrestricted right to read is indeed permitted for a price of \$50. It replies to process A that these terms are available. Process A receives this reply and checks it against the control set in the PERC it uses as

a rule base. The \$50 is outside the \$10 limit specified for this control set, so Process A cannot accept the offer. It makes a counter offer (as described in another optional rights option) of an unrestricted right to read coupled with the release of the reader's name and address. The name and address fields are described in a DTD referenced by Process A's PERC. Process B checks its rules PERC and determines that an unrestricted right to read combined with the release of personal information is a permitted option. It compares the fields that would be released as described in the DTD provided by Process A against the desired fields in a DTD in its own PERC, and determines an acceptable match has occurred. It then sends an offer for unrestricted rights with the release of specific information for the cost of \$5.50 to Process A. Process A compares the right, restrictions, and fields against its rule set and determines that \$5.50 is within the range of \$5-\$6 described as acceptable in its rule set. It accepts the offer as made. The offer is sealed by both parties "signing" a new PERC that describes the results of the final negotiation (unrestricted rights, with release of user information, for \$5.50). The new PERC may be used by the owner of Process A to read the content (the book) subject to the described terms and conditions.

### Further Chain of Handling Model

As described in connection with Figure 2, there are four (4) "participant" instances of VDE 100 in one example of a VDE chain of handling and control used, for example, for content distribution. The first of these participant instances, the content creator 102, is manipulated by the publisher, author, rights owner or distributor of a literary property to prepare the information for distribution to the consumer. The second participant instance, VDE rights distributor 106, may distribute rights and may also administer and analyze customers' use of VDE authored information. The third participant instance, content user 112, is operated by users (included end-users and distributors) when they use information. The fourth participant instance, financial clearinghouse 116 enables the VDE related clearinghouse activities. A further participant, a VDE administrator, may provide support to keep VDE 100 operating properly. With appropriate authorizations and Rights Operating System components installed, any VDE electronic appliance 600 can play any or all of these participant roles.

Literary property is one example of raw material for VDE 100. To transfer this raw material into finished goods, the publisher, author, or rights owner uses tools to transform digital information (such as electronic books, databases, computer

software and movies) into protected digital packages called "objects." Only those consumers (or others along the chain of possession such as a redistributor) who receive permission from a distributor 106 can open these packages. VDE packaged content  
5 can be constrained by "rules and control information" provided by content creator 102 and/or content distributor 106—or by other VDE participants in the content's distribution pathway, i.e., normally by participants "closer" to the creation of the VDE secured package than the participant being constrained.

10

Once the content is packaged in an "object," the digital distribution process may begin. Since the information packages themselves are protected, they may be freely distributed on CD-ROM disks, through computer networks, or broadcast through  
15 cable or by airwaves. Informal "out of channel" exchange of protected packages among end-users does not pose a risk to the content property rights. This is because only authorized individuals may use such packages. In fact, such "out of channel" distribution may be encouraged by some content  
20 providers as a marginal cost method of market penetration. Consumers with usage authorizations (e.g., a VISA clearinghouse budget allowing a certain dollar amount of usage) may, for example, be free to license classes of out of channel

VDE protected packages provided to them, for example, by a neighbor.

5 To open a VDE package and make use of its content, an end-user must have permission. Distributors 106 can grant these permissions, and can very flexibly (if permitted by senior control information) limit or otherwise specify the ways in which package contents may be used. Distributors 106 and financial clearinghouses 116 also typically have financial responsibilities  
10 (they may be the same organization in some circumstances if desired). They ensure that any payments required from end-users fulfill their own and any other participant's requirements. This is achieved by auditing usage.

15 Distributors 106 using VDE 100 may include software publishers, database publishers, cable, television, and radio broadcasters, and other distributors of information in electronic form. VDE 100 supports all forms of electronic distribution, including distribution by broadcast or telecommunications, or by  
20 the physical transfer of electronic storage media. It also supports the delivery of content in homogeneous form, seamlessly integrating information from multiple distribution types with separate delivery of permissions, control mechanisms and content.

Distributors 106 and financial clearinghouses 116 may themselves be audited based on secure records of their administrative activities and a chain of reliable, "trusted" processes ensures the integrity of the overall digital distribution process. This allows content owners, for example, to verify that they are receiving appropriate compensation based on actual content usage or other agreed-upon bases.

Since the end-user 112 is the ultimate consumer of content in this example, VDE 100 is designed to provide protected content in a seamless and transparent way—so long as the end-user stays within the limits of the permissions she has received. The activities of end-user 112 can be metered so that an audit can be conducted by distributors 106. The auditing process may be filtered and/or generalized to satisfy user privacy concerns. For example, metered, recorded VDE content and/or appliance usage information may be filtered prior to reporting it to distributor 106 to prevent more information than necessary from being revealed about content user 112 and/or her usage.

20

VDE 100 gives content providers the ability to recreate important aspects of their traditional distribution strategies in electronic form and to innovatively structure new distribution mechanisms appropriate to their individual needs and



circumstances. VDE 100 supports relevant participants in the chain of distribution, and also enables their desired pricing strategies, access and redistribution permissions, usage rules, and related administrative and analysis procedures. The reusable functional primitives of VDE 100 can be flexibly combined by content providers to reflect their respective distribution objectives. As a result, content providers can feed their information into established distribution channels and also create their own personalized distribution channels.

A summary of the roles of the various participants of virtual distribution environment 100 is set forth in the table below:

15	Role	Description
	<b>Traditional Participants</b>	
	Content creator	Packager and initial distributor of digital information
	Content owner	Owner of the digital information.
	Distributors	Provide rights distribution services for budgets and/or content.
20	Auditor	Provides services for processing and reducing usage based audit trails.
	Clearinghouse	Provides intermediate store and forward services for content and audit information. Also, typically provides a platform for other services, including third party financial providers and auditors.

15

Role	Description
Network provider	Provides communication services between sites and other participants.
Financial providers	Provider of third party sources of electronic funds to end-users and distributors. Examples of this class of users are VISA, American Express, or a government.
End Users	Consumers of information.
<b>Other Participants</b>	
Redistributor	Redistributes rights to use content based on chain of handling restrictions from content providers and/or other distributors.
VDE Administrator	Provider of trusted services for support of VDE nodes.
Independent Audit Processor	Provider of services for processing and summarizing audit trail data. Provides anonymity to end-users while maintaining the comprehensive audit capabilities required by the content providers.
Agents	Provides distributed presence for end-users and other VDE participants.

5

10

15

Of these various VDE participants, the "redistributor," "VDE Administrator," "independent audit processor" and "agents" are, in certain respects "new" participants that may have no counterpart in many "traditional" business models. The other VDE participants (i.e., content provider, content owner, distributors, auditor, clearinghouse, network provider and

financial providers) have "traditional" business model counterparts in the sense that traditional distribution models often included non-electronic participants performing some of the same business roles they serve in the virtual distribution environment 100.

VDE distributors 106 may also include "end-users" who provide electronic information to other end-users. For example, Figure 77 shows a further example of a virtual distribution environment 100 chain of handling and control provided by the present invention. As compared to Figure 2, Figure 77 includes a new "client administrator" participant 700. In addition, Figure 77 shows several different content users 112(1), 112(2), . . . , 112(n) that may all be subject to the "jurisdiction" of the client administrator 700. Client administrator 700 may be, for example, a further rights distributor within a corporation or other organization that distributes rights to employees or other organization participant units (such as divisions, departments, networks, and or groups, etc.) subject to organization-specific "rules and control information." The client administrator 700 may fashion rules and control information for distribution, subject to "rules and control" specified by creator 102 and/or distributor 106.

As mentioned above, VDE administrator 116b is a trusted VDE node that supports VDE 100 and keeps it operating properly. In this example, VDE administrator 116b may provide, among others, any of all of the following:

- 5           • VDE appliance initialization services
- VDE appliance reinitialization/update services
- Key management services
- "Hot lists" of "rogue" VDE sites
- Certification authority services
- 10          • Public key registration
- Client participant unit content budgets and other  
            authorizations

15           All participants of VDE 100 have the innate ability to participate in any role. For example, users may gather together existing protected packages, add (create new content) packages of their own, and create new products. They may choose to serve as their own distributor, or delegate this responsibility to others. These capabilities are particularly important in the object  
20           oriented paradigm which is entering the marketplace today. The production of compound objects, object linking and embedding, and other multi-source processes will create a need for these capabilities of VDE 100. The distribution process provided by VDE 100 is symmetrical; any end-user may redistribute

information received to other end-users, provided they possess permission from and follow the rules established by the distribution chain VDE control information governing redistribution. End-users also may, within the same rules and permissions restriction, encapsulate content owned by others within newly published works and distribute these works independently. Royalty payments for the new works may be accessed by the publisher, distributors, or end-users, and may be tracked and electronically collected at any stage of the chain.

10

Independent financial providers can play an important role in VDE 100. The VDE financial provider role is similar to the role played by organizations such as VISA in traditional distribution scenarios. In any distribution model, authorizing payments for use of products or services and auditing usage for consistency and irregularities, is critical. In VDE 100, these are the roles filled by independent financial providers. The independent financial providers may also provide audit services to content providers. Thus, budgets or limits on use, and audits, or records of use, may be processed by (and may also be put in place by) clearinghouses 116, and the clearinghouses may then collect usage payments from users 112. Any VDE user 112 may assign the right to process information or perform services on their behalf to the extend allowed by senior control information.

15

20

The arrangement by which one VDE participant acts on behalf of another is called a "proxy." Audit, distribution, and other important rights may be "proxied" if permitted by the content provider. One special type of "proxy" is the VDE administrator

5 116b. A VDE administrator is an organization (which may be acting also as a financial clearinghouse 116) that has permission to manage (for example, "intervene" to reset) some portion or all of VDE secure subsystem control information for VDE electronic appliances. This administration right may extend only to

10 admitting new appliances to a VDE infrastructure and to recovering "crashed" or otherwise inoperable appliances, and providing periodic VDE updates.

15 **More On Object Creation, Distribution Methods, Budgets, and Audits**

VDE node electronic appliances 600 in the preferred embodiment can have the ability to perform object creation, distribution, audit collection and usage control functions

20 provided by the present invention. Incorporating this range of capabilities within each of many electronic appliances 600 provided by the preferred embodiment is important to a general goal of creating a single (or prominent) standard for electronic transactions metering, control, and billing, that, in its sum of

25 installations, constitutes a secure, trusted, virtual

transaction/distribution management environment. If, generally speaking, certain key functions were generally or frequently missing, at least in general purpose VDE node electronic appliances 600, then a variety of different products and different standards would come forth to satisfy the wide range of applications for electronic transaction/distribution management; a single consistent set of tools and a single "rational," trusted security and commercial distribution environment will not have been put in place to answer the pressing needs of the evolving "electronic highway." Certain forms of certain electronic appliances 600 containing VDE nodes which incorporate embedded dedicated VDE microcontrollers such as certain forms of video cassette players, cable television converters and the like may not necessarily have or need full VDE capabilities.

However, the preferred embodiment provides a number of distributed, disparately located electronic appliances 600 each of which desirably include authoring, distribution, extraction, audit, and audit reduction capabilities, along with object authoring capabilities.

20

The VDE object authoring capabilities provided by the preferred embodiment provides an author, for example, with a variety of menus for incorporating methods in a VDE object 300, including:

- menus for metering and/or billing methods which define how usage of the content portion of a VDE object is to be controlled,
- 5      • menus related to extraction methods for limiting and/or enabling users of a VDE object from extracting information from that object, and may include placing such information in a newly created and/or pre-existing VDE content container,,
- 10      • menus for specifying audit methods—that is, whether or not certain audit information is to be generated and communicated in some secure fashion back to an object provider, object creator, administrator, and/or
- 15      clearinghouse, and
- menus for distribution methods for controlling how an object is distributed, including for example, controlling distribution rights of different participant's "down" a VDE chain of
- 20      content container handling.

The authoring capabilities may also include procedures for distributing administrative budgets, object distribution control keys, and audit control keys to distributors and other VDE participants who are authorized to perform distribution and/or



auditing functions on behalf of the author, distributors, and/or themselves. The authoring capabilities may also include procedures for selecting and distributing distribution methods, audit methods and audit reduction methods, including for  
5 example, securely writing and/or otherwise controlling budgets for object redistribution by distributors to subsequent VDE chain of content handling participants.

The content of an object 300 created by an author may be  
10 generated with the assistance of a VDE aware application program or a non-VDE aware application program. The content of the object created by an author in conjunction with such programs may include text, formatted text, pictures, moving pictures, sounds, computer software, multimedia, electronic  
15 games, electronic training materials, various types of files, and so on, without limitation. The authoring process may encapsulate content generated by the author in an object, encrypt the content with one or more keys, and append one or more methods to define parameters of allowed use and/or  
20 required auditing of use and/or payment for use of the object by users (and/or by authorized users only). The authoring process may also include some or all aspects of distributing the object.

In general, in the preferred embodiment, an author can:

A. Specify what content is to be included in an object.

B. Specify content oriented methods including:

5 Information--typically abstract, promotional, identifying, scheduling, and/or other information related to the content and/or author

10 Content--e.g. list of files and/or other information resources containing content, time variables, etc.

C. Specify control information (typically a collection of methods related to one another by one or more permissions records, including any method defining variables) and any initial authorized user list including, for example:

15 Control information over Access & Extraction

20 Control information over Distribution  
Control information over Audit Processing

A VDE node electronic appliance 600 may, for example, distribute an object on behalf of an object provider if a VDE node

receives from an object provider administrative budget information for distributing the object and associated distribution key information.

5           A VDE node electronic appliance 600 may receive and process audit records on behalf of an object provider if that VDE node receives any necessary administrative budget, audit method, and audit key information (used, for example, to decrypt  
10           audit trails), from the object provider. An auditing-capable VDE electronic appliance 600 may control execution of audit reduction methods. "Audit reduction" in the preferred embodiment is the process of extracting information from audit records and/or processes that an object provider (e.g., any object provider along  
15           a chain of handling of the object) has specified to be reported to an object's distributors, object creators, client administrators, and/or any other user of audit information. This may include, for example, advertisers who may be required to pay for a user's usage of object content. In one embodiment, for example, a clearinghouse can have the ability to "append" budget, audit  
20           method, and/or audit key information to an object or class or other grouping of objects located at a user site or located at an object provider site to ensure that desired audit processes will take place in a "trusted" fashion. A participant in a chain of handling of a VDE content container and/or content container

control information object may act as a "proxy" for another party in a chain of handling of usage auditing information related to usage of object content (for example a clearinghouse, an advertiser, or a party interested in market survey and/or specific customer usage information). This may be done by specifying, for that other party, budget, audit method, and/or key information that may be necessary to ensure audit information is gathered and/or provided to, in a proper manner, said additional party. For example, employing specification information provided by said other party.

### Object Creation and Initial Control Structures

The VDE preferred embodiment object creation and control structure design processes support fundamental configurability of control information. This enables VDE 100 to support a full range of possible content types, distribution pathways, usage control information, auditing requirements, and users and user groups. VDE object creation in the preferred embodiment employs VDE templates whose atomic elements represent at least in part modular control processes. Employing VDE creation software (in the preferred embodiment a GUI programming process) and VDE templates, users may create VDE objects 300 by, for example, partitioning the objects, placing "meta data" (e.g., author's name, creation date, etc.) into them,

and assigning rights associated with them and/or object content to, for example, a publisher and/or content creator. When an object creator runs through this process, she normally will go through a content specification procedure which will request  
5 required data. The content specification process, when satisfied, may proceed by, for example, inserting data into a template and encapsulating the content. In addition, in the preferred embodiment, an object may also automatically register its presence with the local VDE node electronic appliance 600 secure  
10 subsystem, and at least one permissions record 808 may be produced as a result of the interaction of template instructions and atomic methods, as well as one or more pieces of control structure which can include one or more methods, budgets, and/or etc. A registration process may require a budget to be  
15 created for the object. If an object creation process specifies an initial distribution, an administrative object may also be created for distribution. The administrative object may contain one or more permission records 808, other control structures, methods, and/or load modules.

20

Permissions records 808 may specify various control relationships between objects and users. For example, VDE 100 supports both single access (e.g., one-to-one relationship between a user and a right user) and group access (any number of people

may be authorized as a group). A single permissions record 808 can define both single and group access. VDE 100 may provide "sharing," a process that allows multiple users to share a single control budget as a budget. Additional control structure concepts include distribution, redistribution, and audit, the latter supporting meter and budget information reduction and/or transfer. All of these processes are normally securely controlled by one or more VDE secure subsystems.

## 10      **Templates and Classes**

VDE templates, classes, and flexible control structures support frameworks for organizations and individuals that create, modify, market, distribute, redistribute, consume, and otherwise use movies, audio recordings and live performances, magazines, telephony based retail sales, catalogs, computer software, information databases, multimedia, commercial communications, advertisements, market surveys, infomercials, games, CAD/CAM services for numerically controlled machines, and the like. As the context surrounding these classes changes or evolves, the templates provided by the preferred embodiment of the present invention can be modified to meet these changes for broad use, or more focused activities.

VDE 100 authoring may provide three inputs into a create process: Templates, user input and object content. Templates act as a set of control instructions and/or data for object control software which are capable of creating (and/or modifying) VDE objects in a process that interacts with user instructions and provided content to create a VDE object. Templates are usually specifically associated with object creation and/or control structures. Classes represent user groups which can include "natural" groups within an organization, such as department members, specific security clearance levels, etc., or ad hoc lists of individual's and/or VDE nodes.

For example, templates may be represented as text files defining specific structures and/or component assemblies. Templates, with their structures and/or component assemblies may serve as VDE object authoring or object control applications. A creation template may consist of a number of sub-templates, which, at the lowest level, represent an "atomic level" of description of object specification. Templates may present one or more models that describe various aspects of a content object and how the object should be created including employing secure atomic methods that are used to create, alter, and/or destroy permissions records 808 and/or associated budgets, etc.

Templates, classes (including user groups employing an object under group access), and flexible control structures including object "independent" permissions records (permissions that can be associated with a plurality of objects) and structures that support budgeting and auditing as separate VDE processes, help focus the flexible and configurable capabilities inherent within authoring provided by the present invention in the context of specific industries and/or businesses and/or applications. VDE rationalizes and encompasses distribution scenarios currently employed in a wide array of powerful industries (in part through the use of application or industry specific templates). Therefore, it is important to provide a framework of operation and/or structure to allow existing industries and/or applications and/or businesses to manipulate familiar concepts related to content types, distribution approaches, pricing mechanisms, user interactions with content and/or related administrative activities, budgets, and the like.

The VDE templates, classes, and control structures are inherently flexible and configurable to reflect the breadth of information distribution and secure storage requirements, to allow for efficient adaptation into new industries as they evolve, and to reflect the evolution and/or change of an existing industry and/or business, as well as to support one or more groups of



users who may be associated with certain permissions and/or budgets and object types. The flexibility of VDE templates, classes, and basic control structures is enhanced through the use of VDE aggregate and control methods which have a compound, conditional process impact on object control. Taken together, and employed at times with VDE administrative objects and VDE security arrangements and processes, the present invention truly achieves a content control and auditing architecture that can be configured to most any commercial distribution embodiment. Thus, the present invention fully supports the requirements and biases of content providers without forcing them to fit a predefined application model. It allows them to define the rights, control information, and flow of their content (and the return of audit information) through distribution channels.

**Modifying Object Content (Adding, Hiding, Modifying, Removing, and/or Extending)**

Adding new content to objects is an important aspect of authoring provided by the present invention. Providers may wish to allow one or more users to add, hide, modify, remove and/or extend content that they provide. In this way, other users may add value to, alter for a new purpose, maintain, and/or

otherwise change, existing content. The ability to add content to an empty and/or newly created object is important as well.

5 When a provider provides content and accompanying control information, she may elect to add control information that enables and/or limits the addition, modification, hiding and/or deletion of said content. This control information may concern:

- 10                   • the nature and/or location of content that may be added, hidden, modified, and/or deleted;
- portions of content that may be modified, hidden, deleted and/or added to;
- required secure control information over subsequent VDE container content usage in a chain of control and/or locally to added, hidden, and/or modified content;
- 15                   • requirements that provider-specified notices and/or portions of content accompany added, hidden, deleted and/or modified content and/or the fact that said adding, hiding, modification and/or deletion occurred;
- 20                   • secure management of limitations and/or requirements concerning content that may be removed, hidden and/or deleted from content,

including the amount and/or degree of  
addition, hiding, modification and/or deletion  
of content;

5

- providing notice to a provider or providers that  
modification, hiding, addition and/or deletion  
has occurred and/or the nature of said  
occurrence; and

10

- other control information concerned with  
modification, addition, hiding, and/or deleting  
provider content.

15

A provider may use this control information to establish an  
opportunity for other users to add value to and/or maintain  
existing content in a controlled way. For example, a provider of  
software development tools may allow other users to add  
commentary and/or similar and/or complementary tools to their  
provided objects. A provider of movies may allow commentary  
and/or promotional materials to be added to their materials. A  
provider of CAD/CAM specifications to machine tool owners may  
allow other users to modify objects containing instructions  
associated with a specification to improve and/or translate said  
instructions for use with their equipment. A database owner  
may allow other users to add and/or remove records from a

20

provided database object to allow flexibility and/or maintenance of the database.

5 Another benefit of introducing control information is the opportunity for a provider to allow other users to alter content for a new purpose. A provider may allow other users to provide content in a new setting.

10 To attach this control information to content, a provider may be provided with, or if allowed, design and implement, a method or methods for an object that govern addition, hiding, modification and/or deletion of content. Design and implementation of such one or more methods may be performed using VDE software tools in combination with a PPE 650. The  
15 provider may then attach the method(s) to an object and/or provide them separately. A permissions record 808 may include requirements associated with this control information in combination with other control information, or a separate permissions record 808 may be used.

20

An important aspect of adding or modifying content is the choice of encryption/decryption keys and/or other relevant aspects of securing new or altered content. The provider may specify in their method(s) associated with these processes a

technique or techniques to be used for creating and/or selecting the encryption/decryption keys and/or other relevant aspect of securing new and/or altered content. For example, the provider may include a collection of keys, a technique for generating new  
5 keys, a reference to a load module that will generate keys, a protocol for securing content, and/or other similar information.

Another important implication is the management of new keys, if any are created and/or used. A provider may require  
10 that such keys and reference to which keys were used must be transmitted to the provider, or she may allow the keys and/or securing strategy to remain outside a provider's knowledge and/or control. A provider may also choose an intermediate course in which some keys must be transmitted and others may  
15 remain outside her knowledge and/or control.

An additional aspect related to the management of keys is the management of permissions associated with an object resulting from the addition, hiding, modification and/or deletion  
20 of content. A provider may or may not allow a VDE chain of control information user to take some or all of the VDE rules and control information associated with granting permissions to access and/or manipulate VDE managed content and/or rules and control information associated with said resulting object.

For example, a provider may allow a first user to control access to new content in an object, thereby requiring any other user of that portion of content to receive permission from the first user. This may or may not, at the provider's discretion, obviate the need for a user to obtain permission from the provider to access the object at all.

Keys associated with addition, modification, hiding and/or deletion may be stored in an independent permissions record or records 808. Said permissions record(s) 808 may be delivered to a provider or providers and potentially merged with an existing permissions record or records, or may remain solely under the control of the new content provider. The creation and content of an initial permissions record 808 and any control information over the permissions record(s) are controlled by the method(s) associated with activities by a provider. Subsequent modification and/or use of said permission record(s) may involve a provider's method(s), user action, or both. A user's ability to modify and/or use permissions record(s) 808 is dependent on, at least in part, the senior control information associated with the permissions record(s) of a provider.

### Distribution Control information

To enable a broad and flexible commercial transaction environment, providers should have the ability to establish firm control information over a distribution process without unduly limiting the possibilities of subsequent parties in a chain of control. The distribution control information provided by the present invention allow flexible positive control. No provider is required to include any particular control, or use any particular strategy, except as required by senior control information.

Rather, the present invention allows a provider to select from generic control components (which may be provided as a subset of components appropriate to a provider's specific market, for example, as included in and/or directly compatible with, a VDE application) to establish a structure appropriate for a given chain of handling/control. A provider may also establish control information on their control information that enable and limit modifications to their control information by other users.

The administrative systems provided by the present invention generate administrative "events." These "events" correspond to activities initiated by either the system or a user that correspond to potentially protected processes within VDE. These processes include activities such as copying a permissions record, copying a budget, reading an audit trail record, copying a

method, updating a budget, updating a permissions record,  
updating a method, backing up management files, restoring  
management files, and the like. Reading, writing, modifying,  
updating, processing, and/or deleting information from any  
5 portion of any VDE record may be administrative events. An  
administrative event may represent a process that performs one  
or more of the aforementioned activities on one or more portions  
of one or more records.

10 When a VDE electronic appliance 600 encounters an  
administrative event, that event is typically processed in  
conjunction with a VDE PPE 650. As in the case of events  
generally related to access and/or use of content, in most cases  
administrative events are specified by content providers  
15 (including, for example, content creators, distributors, and/or  
client administrators) as an aspect of a control specified for an  
object, group and/or class of objects.

For example, if a user initiates a request to distribute  
20 permission to use a certain object from a desktop computer to a  
notebook computer, one of the administrative events generated  
may be to create a copy of a permissions record that corresponds  
to the object. When this administrative event is detected by ROS  
602, an EVENT method for this type of event may be present. If



an EVENT method is present, there may also be a meter, a billing, and a budget associated with the EVENT method. Metering, billing, and budgeting can allow a provider to enable and limit the copying of a permissions record 808.

5

For example, during the course of processing a control program, a meter, a billing, and a budget and/or audit records may be generated and/or updated. Said audit records may record information concerning circumstances surrounding an administrative event and processing of said event. For example, an audit record may contain a reference to a user and/or system activity that initiated an event, the success or failure of processing said event, the date and/or time, and/or other relevant information.

10

15

Referring to the above example of a user with both a desktop and notebook computer, the provider of a permissions record may require an audit record each time a meter for copying said permissions record is processed. The audit record provides a flexible and configurable control and/or recording environment option for a provider.

20

In some circumstances, it may be desirable for a provider to limit which aspects of a control component may be modified,

updated, and/or deleted. "Atomic element definitions" may be used to limit the applicability of events (and therefore the remainder of a control process, if one exists) to certain "atomic elements" of a control component. For example, if a permissions record 808 is decomposed into "atomic elements" on the fields described in Figure 26, an event processing chain may be limited, for example, to a certain number of modifications of expiration date/time information by specifying only this field in an atomic element definition. In another example, a permissions record 808 may be decomposed into atomic elements based on control sets. In this example, an event chain may be limited to events that act upon certain control sets.

In some circumstances, it may be desirable for a provider to control how administrative processes are performed. The provider may choose to include in distribution records stored in secure database 610 information for use in conjunction with a component assembly 690 that controls and specifies, for example, how processing for a given event in relation to a given method and/or record should be performed. For example, if a provider wishes to allow a user to make copies of a permissions record 808, she may want to alter the permissions record internally. For example, in the earlier example of a user with a desktop and a notebook computer, a provider may allow a user to make copies

of information necessary to enable the notebook computer based on information present in the desktop computer, but not allow any further copies of said information to be made by the notebook VDE node. In this example, the distribution control structure described earlier would continue to exist on the desktop computer, but the copies of the enabling information passed to the notebook computer would lack the required distribution control structure to perform distribution from the notebook computer. Similarly, a distribution control structure may be provided by a content provider to a content provider who is a distributor in which a control structure would enable a certain number of copies to be made of a VDE content container object along with associated copies of permissions records, but the permissions records would be altered (as per specification of the content provider, for example) so as not to allow end-users who received distributor created copies from making further copies for distribution to other VDE nodes.

Although the preceding example focuses on one particular event (copying) under one possible case, similar processes may be used for reading, writing, modifying, updating, processing, and/or deleting information from records and/or methods under any control relationship contemplated by the present invention. Other examples include: copying a budget, copying a meter,

updating a budget, updating a meter, condensing an audit trail, and the like.

#### **Creating Custom Methods**

5           In the preferred embodiment of the present invention, methods may be created "at will," or aliased to another method. These two modes contribute to the superior configurability, flexibility, and positive control of the VDE distribution process. Generally, creating a method involves specifying the required  
10       attributes or parameters for the data portion of the method, and then "typing" the method. The typing process typically involves choosing one or more load modules to process any data portions of a method. In addition to the method itself, the process of method creation may also result in a method option subrecord for  
15       inclusion in, or modification of, a permissions record, and a notation in the distribution records. In addition to any "standard" load module(s) required for exercise of the method, additional load modules, and data for use with those load modules, may be specified if allowed. These event processing  
20       structures control the distribution of the method.

For example, consider the case of a security budget. One form of a typical budget might limit the user to 10Mb of decrypted data per month. The user wishes to move their rights

to use the relevant VDE content container object to their notebook. The budget creator might have limited the notebook to the same amount, half the original amount, a prorated amount based on the number of moves budgeted for an object, etc. A  
5 distribute method (or internal event processing structure) associated with the budget allows the creator of the budget to make a determination as to the methodology and parameters involved. Of course, different distribution methods may be required for the case of redistribution, or formal distribution of  
10 the method. The aggregate of these choices is stored in a permissions record for the method.

An example of the process steps used for the move of a budget record might look something like this:

- 15 1) Check the move budget (e.g., to determine the number of moves allowed)
- 2) Copy static fields to new record (e.g., as an encumbrance)
- 3) Decrement the Decr counter in the old record (the  
20 original budget)
- 4) Increment the Encumbrance counter in the old record
- 5) Write a distribution record
- 6) Write a Distribution Event Id to the new record

- 7) Increment the move meter
- 8) Decrement the move budget
- 9) Increment the Decr counter in the new record

## 5      **Creating a Budget**

In the preferred embodiment, to create a budget, a user manipulates a Graphical User Interface budget distribution application (e.g., a VDE template application). The user fills out any required fields for type(s) of budget, expiration cycle(s),  
10      auditor(s), etc. A budget may be specified in dollars, deutsche marks, yen, and/or in any other monetary or content measurement schema and/or organization. The preferred embodiment output of the application, normally has three basic elements. A notation in the distribution portion of secure  
15      database 610 for each budget record created, the actual budget records, and a method option record for inclusion in a permissions record. Under some circumstances, a budget process may not result in the creation of a method option since an existing method option may be being used. Normally, all of this  
20      output is protected by storage in secure database 610 and/or in one or more administrative objects.

There are two basic modes of operation for a budget distribution application in the preferred embodiment. In the

first case, the operator has an unlimited ability to specify budgets. The budgets resulting from this type of activity may be freely used to control any aspect of a distribution process for which an operator has rights, including for use with "security" budgets such as quantities limiting some aspect of usage. For example, if the operator is a "regular person," he may use these budgets to control his own utilization of objects based on a personal accounting model or schedule. If the operator is an authorized user at VISA, the resulting budgets may have broad implications for an entire distribution system. A core idea is that this mode is controlled strictly by an operator.

The second mode of operation is used to create "alias" budgets. These budgets are coupled to a preexisting budget in an operator's system. When an operator fills a budget, an encumbrance is created on the aliased budget. When these types of budgets are created, the output includes two method option subrecords coupled together: the method option subrecord for the aliased budget, and a method option subrecord for the newly created budget. In most cases, the alias budget can be used in place of the original budget if the budget creator is authorized to modify the method options within the appropriate required method record of a permissions record.

For example, assume that a user (client administrator) at a company has the company's VISA budget on her electronic appliance 600. She wants to distribute budget to a network of company users with a variety of preexisting budgets and requirements. She also wants to limit use of the company's VISA budget to certain objects. To do this, she aliases a company budget to the VISA budget. She then modifies (if so authorized) the permissions record for all objects that the company will allow their users to manipulate so that they recognize the company budget in addition to, or instead of, the VISA budget. She then distributes the new permissions records and budgets to her users. The audit data from these users is then reduced against the encumbrance on the company's VISA budget to produce a periodic billing.

15

In another example, a consumer wants to control his family's electronic appliance use of his VISA card, and prevent his children from playing too many video games, while allowing unlimited use of encyclopedias. In this case, he could create two budgets. The first budget can be aliased to his VISA card, and might only be used with encyclopedia objects (referenced to individual encyclopedia objects and/or to one or more classes of encyclopedia objects) that reference the aliased budget in their explicitly modified permissions record. The second budget could

20



be, for example, a time budget that he redistributes to the family for use with video game objects (video game class). In this instance, the second budget is a "self-replenishing" security/control budget, that allows, for example, two hours of use per day. The first budget operates in the same manner as the earlier example. The second budget is added as a new required method to permissions records for video games. Since the time budget is required to access the video games, an effective control path is introduced for requiring the second budget -- only permissions records modified to accept the family budget can be used by the children for video games and they are limited to two hours per day.

### Sharing and Distributing Rights and Budgets

#### 15           Move

          The VDE "move" concept provided by the preferred embodiment covers the case of "friendly sharing" of rights and budgets. A typical case of "move" is a user who owns several machines and wishes to use the same objects on more than one of them. For example, a user owns a desktop and a notebook computer. They have a subscription to an electronic newspaper that they wish to read on either machine, i.e., the user wishes to move rights from one machine to the other.

An important concept within "move" is the idea of independent operation. Any electronic appliance 600 to which rights have been moved may contact distributors or clearinghouses independently. For example, the user mentioned  
5 above may want to take their notebook on the road for an extended period of time, and contact clearinghouses and distributors without a local connection to their desktop.

To support independent operation, the user should be able  
10 to define an account with a distributor or clearinghouse that is independent of the electronic appliance 600 she is using to connect. The transactions must be independently traceable and reconcilable among and between machines for both the end user and the clearinghouse or distributor. The basic operations of  
15 moving rights, budgets, and bitmap or compound meters between machines is also supported.

---

### **Redistribution**

Redistribution forms a UDE middle ground between the  
20 "friendly sharing" of "move," and formal distribution. Redistribution can be thought of as "anonymous distribution" in the sense that no special interaction is required between a creator, clearinghouse, or distributor and a redistributor. Of

course, a creator or distributor does have the ability to limit or prevent redistribution.

5 Unlike the "move" concept, redistribution does not imply independent operation. The redistributor serves as one point of contact for users receiving redistributed rights and/or budgets, etc. These users have no knowledge of, or access to, the clearinghouse (or and/or distributor) accounts of the redistributor. The redistributor serves as an auditor for the  
10 rights and/or budgets, etc. that they redistribute, unless specifically overridden by restrictions from distributors and/or clearinghouses. Since redistributees (recipients of redistributed rights and/or budgets, etc.) would place a relatively unquantifiable workload on clearinghouses. and furthermore,  
15 since a redistributor would be placing himself at an auditable risk (responsible for all redistributed rights and/or budgets, etc.), the audit of rights, budgets, etc. of redistributees by redistributors is assumed as the default case in the preferred embodiment.

20

### **Distribution**

Distribution involves three types of entity. Creators usually are the source of distribution. They typically set the control structure "context" and can control the rights which are

passed into a distribution network. Distributors are users who  
form a link between object (content) end users and object  
(content) creators. They can provide a two-way conduit for rights  
and audit data. Clearinghouses may provide independent  
5 financial services, such as credit and/or billing services, and can  
serve as distributors and/or creators. Through a permissions and  
budgeting process, these parties collectively can establish fine  
control over the type and extent of rights usage and/or auditing  
activities.

10

### **Encumbrance**

An "encumbrance" is a special type of VDE budget. When  
that a budget distribution of any type occurs, an "encumbrance"  
may be generated. An encumbrance is indistinguishable from an  
15 original budget for right exercise (e.g., content usage payment)  
purposes, but is uniquely identified within distribution records  
as to the amount of the encumbrance, and all necessary  
information to complete a shipping record to track the  
whereabouts of an encumbrance. For right exercise purposes, an  
20 encumbrance is identical to an original budget; but for tracking  
purposes, it is uniquely identifiable.

In the preferred embodiment of the present invention, a  
Distribution Event ID will be used by user VDE nodes and by

clearinghouse services to track and reconcile encumbrances, even in the case of asynchronous audits. That is, the "new" encumbrance budget is unique from a tracking point of view, but indistinguishable from a usage point of view.

5

Unresolved encumbrances are a good intermediate control for a VDE distribution process. A suitable "grace period" can be introduced during which encumbrances must be resolved. If this period elapses, an actual billing or payment may occur.

10

However, even after the interval has expired and the billing and/or payment made, an encumbrance may still be outstanding and support later reconciliation. In this case, an auditor may allow a user to gain a credit, or a user may connect to a VDE node containing an encumbered budget, and resolve an amount as an internal credit. In some cases, missing audit trails may concern a distributor sufficiently to revoke redistribution privileges if encumbrances are not resolved within a "grace period," or if there are repeated grace period violations or if unresolved encumbrances are excessively large.

15

20

Encumbrances can be used across a wide variety of distribution modes. Encumbrances, when used in concert with aliasing of budgets, opens important additional distribution possibilities. In the case of aliasing a budget, the user places

himself in the control path for an object -- an aliased budget may only be used in conjunction with permissions records that have been modified to recognize it. An encumbrance has no such restrictions.

5

For example, a user may want to restrict his children's use of his electronic, VDE node VISA budget. In this case, the user can generate an encumbrance on his VISA budget for the children's family alias budget, and another for his wife that is a transparent encumbrance of the original VISA budget. BigCo may use a similar mechanism to distribute VISA budget to department heads, and aliased BigCo budget to users directly.

10

#### **Account Numbers and User IDs**

15

In the preferred embodiment, to control access to clearinghouses, users are assigned account numbers at clearinghouses. Account numbers provide a unique "instance" value for a secure database record from the point of view of an outsider. From the point of view of an electronic appliance 600 site, the user, group, or group/user ids provide the unique instance of a record. For example, from the point of view of VISA, your Gold Card belongs to account number #123456789. From the point of view of the electronic appliance site (for example, a server at a corporation), the Gold card might belong

20

to user id 1023. In organizations which have plural users and/or user groups using a VDE node, such users and/or user groups will likely be assigned unique user IDs. differing budgets and/or other user rights may be assigned to different users and/or user groups and/or other VDE control information may be applied on a differing manner to electronic content and/or appliance usage by users assigned with different such IDs. Of course, both a clearinghouse and a local site will likely have both pieces of information, but "used data" versus the "comment data" may differ based on perspective.

In the preferred embodiment case of "move," an account number stored with rights stays the same. In the preferred embodiment of other forms of distribution, a new account number is required for a distributee. This may be generated automatically by the system, or correspond to a methodology developed by a distributor or redistributor. Distributors maintain account numbers (and associated access secrets) in their local name services for each distributee. Conversely, distributees' name services may store account numbers based on user id for each distributor. This record usually is moved with other records in the case of move, or is generated during other forms of distribution.

Organizations (including families) may automatically assign unique user IDs when creating control information (e.g., a budget) for a new user or user group.

5                   **Requirements Record**

In order to establish the requirements, and potentially options, for exercising a right associated with a VDE content container object before one or more required permissions records are received for that object, a requirements record may exist in the private header of such an object. This record will help the user establish what they have, and what they need from a distributor prior to forming a connection. If the requirements or possibilities for exercising a particular right have changed since such an object was published, a modified requirements record may be included in a container with an object (if available and allowed), or a new requirements record may be requested from a distributor before registration is initiated. Distributors may maintain "catalogs" online, and/or delivered to users, of collections of requirements records and/or descriptive information corresponding to objects for which they may have ability to obtain and/or grant rights to other users.

10

15

20



### Passing an Audit

5 In the preferred embodiment of VDE there may be at least two types of auditing. In the case of budget distribution, billing records that reflect consumption of a budget generally need to be collected and processed. In the case of permissions distribution, usage data associated with an object are also frequently required.

10 In order to effect control over an object, a creator may establish the basic control information associated with an object. This is done in the formulation of permissions, the distribution of various security, administrative and/or financial budgets, and the level of redistribution that is allowed, etc. Distributors (and redistributors) may further control this process within the rights, budgets, etc. (senior control information) they have received.

15 For example, an object creator may specify that additional required methods may be added freely to their permissions records, establish no budget for this activity, and allow unlimited redistribution of this right. As an alternative example, a creator may allow moving of usage rights by a distributor to half a dozen subdistributors, each of whom can distribute 10,000 copies, but with no redistribution rights being allowed to be allocated to subdistributors' (redistributors') customers. As another example, a creator may authorize the moving of usage rights to only 10

VDE nodes, and to only one level of distribution (no redistribution). Content providers and other contributors of control information have the ability through the use of permissions records and/or component assemblies to control rights other users are authorized to delegate in the permissions records they send to those users, so long as such right to control one, some, or all such rights of other users is either permitted or restricted (depending on the control information distribution model). It is possible and often desirable, using VDE, to construct a mixed model in which a distributor is restricted from controlling certain rights of subsequent users and is allowed to control other rights. VDE control of rights distribution in some VDE models will in part or whole, at least for certain one or more "levels" of a distribution chain, be controlled by electronic content control information providers who are either not also providers of the related content or provide only a portion of the content controlled by said content control information. for example, in certain models, a clearinghouse might also serve as a rights distribution agent who provides one or more rights to certain value chain participants, which one or more rights may be "attached" to one or more rights to use the clearinghouse's credit (if said clearinghouse is, at least in part, a financial clearinghouse (such a control information provider may alternatively, or in addition, restrict other users' rights.

A content creator or other content control information provider may budget a user (such as a distributor) to create an unlimited number of permissions records for a content object, but revoke this right and/or other important usage rights through an expiration/termination process if the user does not report his usage (provide an audit report) at some expected one or more points in time and/or after a certain interval of time (and/or if the user fails to pay for his usage or violates other aspects of the agreement between the user and the content provider). This termination (or suspension or other specified consequence) can be enforced, for example, by the expiration of time-aged encryption keys which were employed to encrypt one or more aspects of control information. This same termination (or other specified consequence such as budget reduction, price increase, message displays on screen to users, messages to administrators, etc.) can also be the consequence of the failure by a user or the users VDE installation to complete a monitored process, such as paying for usage in electronic currency, failure to perform backups of important stored information (e.g., content and/or appliance usage information, control information, etc.), failure to use a repeated failure to use the proper passwords or other identifiers, etc.).

Generally, the collection of audit information that is collected for reporting to a certain auditor can be enforced by expiration and/or other termination processes. For example, the user's VDE node may be instructed (a) from an external source to no longer perform certain tasks, (b) carries within its control structure information informing it to no longer perform certain tasks, or (c) is otherwise no longer able to perform certain tasks. The certain tasks might comprise one or more enabling operations due to a user's (or installation's) failure to either report said audit information to said auditor and/or receive back a secure confirmation of receipt and/or acceptance of said audit information. If an auditor fails to receive audit information from a user (or some other event fails to occur or occur properly), one or more time-aged keys which are used, for example, as a security component of an embodiment of the present invention, may have their aging suddenly accelerated (completed) so that one or more processes related to said time-aged keys can no longer be performed.

#### **20 Authorization Access Tags and Modification Access Tags**

In order to enable a user VDE installation to pass audit information to a VDE auditing party such as a Clearinghouse, VDE allows a VDE auditing party to securely, electronically communicate with the user VDE installation and to query said

installation for certain or all information stored within said installation's secure sub-system, depending on said auditing party's rights (said party shall normally be unable to access securely stored information that said party is not expressly  
5 authorized to access, that is one content provider will normally not be authorized to access content usage information related to content provided by a different content provider). The auditing party asserts a secure secret (e.g., a secure tag) that represents the set of rights of the auditor to access certain information  
10 maintained by said subsystem. If said subsystem validates said tag, the auditing party may then receive auditing information that it is allowed to request and receive.

Great flexibility exists in the enforcement of audit trail  
15 requirements. For example, a creator (or other content provider or control information provider or auditor in an object's or audit report's chain of handling) may allow changes by an auditor for event trails, but not allow anyone but themselves to read those trails, and limit the redistribution of this right to, for example,  
20 six levels. Alternatively, a creator or other controlling party may give a distributor the right to process, for example, 100,000 audit records (and/or, for example, the right to process 12 audit records from a given user) before reporting their usage. If a creator or other controlling party desires, he may allow (and/or require)

5 separate (and containing different, a subset of, overlapping, or  
the same information) audit "packets" containing audit  
information, certain of said audit information to be processed by  
a distributor and certain other of said audit information to be  
10 passed back to the creator and/or other auditors (each receiving  
the same, overlapping, a subset of, or different audit  
information). Similarly, as long as allowed by, for example, an  
object creator, a distributor (or other content and/or control  
information provider) may require audit information to be passed  
15 back to it, for example, after every 50,000 audit records are  
processed (or any other unit of quantity and/or after a certain  
time interval and/or at a certain predetermined date) by a  
redistributor. In the preferred embodiment, audit rules, like  
other control structures, may be stipulated at any stage of a  
20 distribution chain of handling as long as the right to stipulate  
said rules has not been restricted by a more "senior" object  
and/or control information distributing (such as an auditing)  
participant.

20           Audit information that is destined for different auditors  
may be encrypted by different one or more encryption keys which  
have been securely provided by each auditor's VDE node and  
communicated for inclusion in a user's permissions record(s) as a  
required step, for example, during object registration. This can

provide additional security to further ensure (beyond the use of passwords and/or other identification information and other VDE security features) that an auditor may only access audit information to which he is authorized. In one embodiment, 5 encrypted (and/or unencrypted) "packets" of audit information (for example, in the form of administrative objects) may be bound for different auditors including a clearinghouse and/or content providers and/or other audit information users (including, for example, market analysts and/or list purveyors). The 10 information may pass successively through a single chain of handling, for example, user to clearinghouse to redistributor to distributor to publisher/object creator, as specified by VDE audit control structures and parameters. Alternatively, encrypted (or, normally less preferably, unencrypted) audit packets may be 15 required to be dispersed directly from a user to a plurality of auditors, some one or more who may have the responsibility to "pass along" audit packets to other auditors. In another embodiment, audit information may be passed, for example, to a clearinghouse, which may then redistribute all and/or 20 appropriate subsets of said information (and/or some processed result) to one or more other parties, said redistribution employing VDE secure objects created by said clearinghouse.

An important function of an auditor (receiver of audit information) is to pass administrative events back to a user VDE node in acknowledgement that audit information has been received and/or "recognized." In the preferred embodiment, the receipt and/or acceptance of audit information may be followed by two processes. The first event will cause the audit data at a VDE node which prepared an audit report to be deleted, or compressed into, or added to, one or more summary values. The second event, or set of events, will "inform" the relevant security (for example, termination and/or other consequence) control information (for example, budgets) at said VDE node of the audit receipt, modify expiration dates, provide key updates, and/or etc. In most cases, these events will be sent immediately to a site after an audit trail is received. In some cases, this transmission may be delayed to, for example, first allow processing of the audit trail and/or payment by a user to an auditor or other party.

In the preferred embodiment, the administrative events for content objects and independently distributed methods/component assemblies are similar, but not necessarily identical. For example, key updates for a budget may control encryption of a billing trail, rather than decryption of object content. The billing trail for a budget is in all respects a method event trail. In one embodiment, this trail must include sufficient



references into distribution records for encumbrances to allow  
reconciliation by a clearinghouse. This may occur, for example, if  
a grace period elapses and the creator of a budget allows  
unresolved encumbrances to ultimately yield automatic credits if  
5 an expired encumbrance is "returned" to the creator.

Delivery of audit reports through a path of handling may  
be in part insured by an inverse (return of information) audit  
method. Many VDE methods have at least two pieces: a portion  
10 that manages the process of producing audit information at a  
user's VDE node; and a portion that subsequently acts on audit  
data. In an example of the handling of audit information bound  
for a plurality of auditors, a single container object is received at  
a clearinghouse (or other auditor). This container may contain  
15 (a) certain encrypted audit information that is for the use of the  
clearinghouse itself, and (b) certain other encrypted audit  
information bound for other one or more auditor parties. The  
two sets of information may have the same, overlapping and in  
part different, or entirely different, information content.  
20 Alternatively, the clearinghouse VDE node may be able to work  
with some or all of the provided audit information. The audit  
information may be, in part, or whole, in some summary and/or  
analyzed form further processed at the clearinghouse and/or may  
be combined with other information to form a, at least in part,

derived set of information and inserted into one or more at least  
in part secure VDE objects to be communicated to said one or  
more (further) auditor parties. When an audit information  
container is securely processed at said clearinghouse VDE node  
5 by said inverse (return) audit method, the clearinghouse VDE  
node can create one or more VDE administrative objects for  
securely carrying audit information to other auditors while  
separately processing the secure audit information that is  
specified for use by said clearinghouse. Secure audit processes  
10 and credit information distribution between VDE participants  
normally takes place within the secure VDE "black box," that is  
processes are securely processed within secure VDE PPE650 and  
audit information is securely communicated between the VDE  
secure subsystems of vDE participants employing VDE secure  
15 communication techniques (e.g., public key encryption, and  
authentication).

This type of inverse audit method may specify the  
handling of returned audit information, including, for example,  
20 the local processing of audit information and/or the secure  
passing along of audit information to one or more auditor parties.  
If audit information is not passed to one or more other auditor  
parties as may be required and according to criteria that may  
have been set by said one or more other auditor parties and/or

content providers and/or control information providers during a permissions record specification and/or modification process, the failure to, for example, receive notification of successful transfer of required audit information by an auditor party, e.g., a content  
5 provider, can result in the disablement of at least some capability of the passing through party's VDE node (for example, disablement of the ability to further perform certain one or more VDE managed business functions that are related to object(s) associated with said audit or party). In this preferred  
10 embodiment example, when an object is received by an auditor, it is automatically registered and permissions record(s) contents are entered into the secure management database of the auditor's VDE node.

15 One or more permissions records that manage the creation and use of an audit report object (and may manage other aspects of object use as well) may be received by a user's system during an audit information reporting exchange (or other electronic interaction between a user and an auditor or auditor agent).  
20 Each received permissions record may govern the creation of the next audit report object. After the reporting of audit information, a new permissions record may be required at a user's VDE node to refresh the capability of managing audit report creation and audit information transfer for the next audit

reporting cycle. In our above example, enabling an auditor to supply one or more permissions records to a user for the purpose of audit reporting may require that an auditor (such as a clearinghouse) has received certain, specified permissions records itself from "upstream" auditors (such as, for example, content and/or other content control information providers). Information provided by these upstream permissions records may be integrated into the one or more permissions records at an auditor VDE (e.g., clearinghouse) installation that manage the permissions record creation cycle for producing administrative objects containing permissions records that are bound for users during the audit information reporting exchange. If an upstream auditor fails to receive, and/or is unable to process, required audit information, this upstream auditor may fail to provide to the clearinghouse (in this example) the required permissions record information which enables a distributor to support the next permission record creation/auditing cycle for a given one or more objects (or class of objects). As a result, the clearinghouse's VDE node may be unable to produce the next cycle's permissions records for users, and/or perform some other important process. This VDE audit reporting control process may be entirely electronic process management involving event driven VDE activities at both the intended audit information receiver and

sender and employing both their secure PPE650 and secure VDE communication techniques.

5 In the preferred embodiment, each time a user registers a new object with her own VDE node, and/or alternatively, with a remote clearinghouse and/or distributor VDE node, one or more permissions records are provided to, at least in part, govern the use of said object. The permissions records may be provided dynamically during a secure UDE registration process

10 (employing the VDE installation secure subsystem), and/or may be provided following an initial registration and received at some subsequent time, e.g. through one or more separate secure VDE communications, including, for example, the receipt of a physical arrangement containing or otherwise carrying said information.

15 At least one process related to the providing of the one or more permissions records to a user can trigger a metering event which results in audit information being created reflecting the user's VDE node's, clearinghouse's, and/or distributor's permissions records provision process. This metering process may not only

20 record that one or more permissions records have been created. It may also record the VDE node name, user name, associated object identification information, time, date, and/or other identification information. Some or all of this information can become part of audit information securely reported by a

clearinghouse or distributor, for example, to an auditing content creator and/or other content provider. This information can be reconciled by secure VDE applications software at a receiving auditor's site against a user's audit information passed through  
5 by said clearinghouse or distributor to said auditor. For each metered one or more permissions records (or set of records) that were created for a certain user (and/or VDE node) to manage use of certain one or more VDE object(s) and/or to manage the creation of VDE object audit reports, it may be desirable that an  
10 auditor receive corresponding audit information incorporated into an, at least in part, encrypted audit report. This combination of metering of the creation of permissions records; secure, encrypted audit information reporting processes; secure VDE subsystem reconciliation of metering information reflecting  
15 the creation of registration and/or audit reporting permissions with received audit report detail; and one or more secure VDE installation expiration and/or other termination and/or other consequence processes; taken together significantly enhances the integrity of the VDE secure audit reporting process as a trusted,  
20 efficient, commercial environment.

### Secure Document Management Example

VDE 100 may be used to implement a secure document management environment. The following are some examples of how this can be accomplished.

5

In one example, suppose a law firm wants to use VDE 100 to manage documents. In this example, a law firm that is part of a litigation team might use VDE in the following ways:

10

1. to securely control access to, and/or other usage of, confidential client records,
2. to securely control access, distribution, and/or other rights to documents and memoranda created at the law firm.

15

3. to securely control access and other use of research materials associated with the case,
4. to securely control access and other use, including distribution of records, documents, and notes associated with the case,

20

5. to securely control how other firms in the litigation team may use, including change, briefs that have been distributed for comment and review,
6. to help manage client billing.

5 The law firm may also use VDE to electronically file briefs with the court (presuming the court is also VDE capable) including providing secure audit verification of the ID (e.g., digital signature) of filers and other information pertinent to said filing procedure.

10 In this example, the law firm receives in VDE content containers documents from their client's VDE installation secure subsystem(s). Alternatively, or in addition, the law firm may receive either physical documents which may be scanned into electronic form, and/or they receive electronic documents which have not yet been placed in VDE containers. The electronic form of a document is stored as a VDE container (object) associated with the specific client and/or case. The VDE container  
15 mechanism supports a hierarchical ordering scheme for organizing files and other information within a container; this mechanism may be used to organize the electronic copies of the documents within a container. A VDE container is associated with specific access control information and rights that are  
20 described in one or more permissions control information sets (PERCs) associated with that container. In this example, only those members of the law firm who possess a VDE instance, an appropriate PERC, and the VDE object that contains the desired document, may use the document. Alternatively or in addition, a



law firm member may use a VDE instance which has been installed on the law firm's network server. In this case, the member must be identified by an appropriate PERC and have access to the document containing VDE object (in order to use the server VDE installation). Basic access control to electronic documents is enabled using the secure subsystem of one or more user VDE installations.

VDE may be used to provide basic usage control in several ways. First, it permits the "embedding" of multiple containers within a single object. Embedded objects permit the "nesting" of control structures within a container. VDE also extends usage control information to an arbitrary granular level (as opposed to a file based level provided by traditional operating systems) and provides flexible control information over any action associated with the information which can be described as a VDE controlled process. For example, simple control information may be associated with viewing the one or more portions of documents and additional control information may be associated with editing, printing and copying the same and/or different one or more portions of these same documents.

In this example, a "client" container contains all documents that have been provided by the client (documents

received in other containers can be securely extracted and embedded into the VDE client container using VDE extraction and embedding capabilities). Each document in this example is stored as an object within the parent, client VDE container. The

5 "client" container also has several other objects embedded within it; one for each attorney to store their client notes, one (or more) for research results and related information, and at least one for copies of letters, work papers, and briefs that have been created by the law firm. The client container may also contain other

10 information about the client, including electronic records of billing, time, accounting, and payments. Embedding VDE objects within a parent VDE content container provides a convenient way to securely categorize and/or store different information that shares similar control information. All client

15 provided documents may, for example, be subject to the same control structures related to use and non-disclosure. Attorney notes may be subject to control information, for example, their use may be limited to the attorney who created the notes and those attorneys to whom such creating attorney expressly grants

20 access rights. Embedded containers also provide a convenient mechanism to control collections of dissimilar information. For example, the research object(s) may be stored in the form of (or were derived from) VDE "smart objects" that contain the results of research performed by that object. Research results related to

one aspect of the case retrieved from a VDE enabled LEXIS site might be encapsulated as one smart object; the results of another session related to another (or the same) aspect of the case may be encapsulated as a different object. Smart objects are used in this  
5 example to help show that completely disparate and separately delivered control information may be incorporated into a client container as desired and/or required to enforce the rights of providers (such as content owners).

10 Control structures may be employed to manage any variety of desired granularities and/or logical document content groupings; a document, page, paragraph, topically related materials, etc. In this example, the following assumptions are made: client provided documents are controlled at the page  
15 level, attorney notes are controlled at the document level on an attorney by attorney basis, court filings and briefs are controlled at a document level, research information is controlled at whatever level the content provider specifies at the time the research was performed, and certain highly confidential  
20 information located in various of the above content may be identified as subject to display and adding comments only, and only by the lead partner attorneys, with only the creator and/or embedder of a given piece of content having the right to be otherwise used (printed, extracted, distributed, etc).

In general, container content in this example is controlled with respect to distribution of rights. This control information are associated at a document level for all internally generated documents, at a page level for client level documents, and at the level specified by the content provider for research documents.

VDE control information can be structured in either complex or simple structures, depending on the participant's desires. In some cases, a VDE creator will apply a series of control structure definitions that they prefer to use (and that are supported by the VDE application managing the specification of rules and control information, either directly, or through the use of certified application compatible VDE component assemblies.

In this example, the law firm sets up a standard VDE client content container for a new client at the time they accept the case. A law firm VDE administrator would establish a VDE group for the new client and add the VDE IDs of the attorneys at the firm that are authorized to work on the case, as well as provide, if appropriate, one or more user template applications. These templates provide, for example, one or more user interfaces and associated control structures for selection by users of additional and/or alternative control functions (if allowed by senior control information), entry of control parameter data,

and/or performing user specific administrative tasks. The administrator uses a creation tool along with a predefined creation template to create the container. This creation template specifies the document usage (including distribution control information) for documents as described above. Each electronic document from the client (including letters, memoranda, E-mail, spreadsheet, etc.) are then added to the container as separate embedded objects. Each new object is created using a creation template that satisfies that the default control structures specified with the container as required for each new object of a given type.

As each attorney works on the case, they may enter notes into an object stored within the client's VDE container. These notes may be taken using a VDE aware word processor already in use at the law firm. In this example, a VDE redirector handles the secure mapping of the word processor file requests into the VDE container and its objects through the use of VDE control processes operating with one or more VDE PPEs. Attorney note objects are created using the default creation template for the document type with assistance from the attorney if the type cannot be automatically determined from the content. This permits VDE to automatically detect and protect

the notes at the predetermined level, e.g. document, page, paragraph.

Research can be automatically managed using VDE.

- 5 Smart objects can be, used to securely search out, pay for if necessary, and retrieve information from VDE enabled information resources on the information highway.

- 10 Examples of such resources might include LEXIS, Westlaw, and other related legal databases. Once the information is retrieved, it may be securely embedded in the VDE content client container. If the smart object still contains unreleased information, the entire smart object may be embedded in the client's VDE container. This places the
- 15 unreleased information under double VDE control requirements: those associated with releasing the information from smart object (such as payment and/or auditing requirements) and those associated with access to, or other usage of, client information of the specified type.

20

Briefs and other filings may be controlled in a manner similar to that for attorney notes. The filings may be edited using the standard word processors in the law firm; with usage control structures controlling who may review, change, and/or

add to the document (or, in a more sophisticated example, a certain portion of said document). VDE may also support electronic filing of briefs by providing a trusted source for time/date stamping and validation of filed documents.

5

When the client and attorney want to exchange confidential information over electronic mail or other means, VDE can play an important role in ensuring that information exchanged under privilege, properly controlled, and not  
10 inappropriately released and/or otherwise used. The materials (content) stored in a VDE content container object will normally be encrypted. Thus wrapped, a VDE object may be distributed to the recipient without fear of unauthorized access and/or other use. The one or more authorized users who have received an  
15 object are the only parties who may open that object and view and/or manipulate and/or otherwise modify its contents and VDE secure auditing ensures a record of all such user content activities. VDE also permits the revocation of rights to use client/attorney privileged information if such action becomes  
20 necessary, for example, after an administrator review of user usage audit information.

### Large Organization Example

In a somewhat more general example, suppose an organization (e.g., a corporation or government department) with thousands of employees and numerous offices disposed throughout a large geographic area wishes to exercise control over distribution of information which belongs to said organization (or association). This information may take the form of formal documents, electronic mail messages, text files, multimedia files, etc., which collectively are referred to as "documents."

Such documents may be handled by people (referred to as "users") and/or by computers operating on behalf of users. The documents may exist both in electronic form for storage and transmission and in paper form for manual handling.

These documents may originate wholly within the organization, or may be created, in whole or in part, from information received from outside the organization. Authorized persons within the organization may choose to release documents, in whole or in part, to entities outside the organization. Some such entities may also employ VDE 100 for document control, whereas others may not.



## Document Control Policies

5 The organization as a whole may have a well-defined policy for access control to, and/or other usage control of documents. This policy may be based on a "lattice model" of information flow, in which documents are characterized as having one or more hierarchical "classification" security attributes 9903 and zero or more non-hierarchical "compartment" security attributes, all of which together comprise a sensitivity security attribute.

10 The classification attributes may designate the overall level of sensitivity of the document as an element of an ordered set. For example, the set "unclassified," "confidential," "secret," "top secret" might be appropriate in a government setting, and  
15 the set "public," "internal," "confidential," "registered confidential" might be appropriate in a corporate setting.

20 The compartment attributes may designate the document's association with one or more specific activities within the organization, such as departmental subdivisions (e.g., "research," "development," "marketing") or specific projects within the organization.

Each person using an electronic appliance 600 would be assigned, by an authorized user, a set of permitted sensitivity attributes to designate those documents, or one or more portions of certain document types, which could be processed in certain one or more ways, by the person's electronic appliance. A document's sensitivity attribute would have to belong to the user's set of permitted sensitivity values to be accessible.

In addition, the organization may desire to permit users to exercise control over specific documents for which the user has some defined responsibility. As an example, a user (the "originating user") may wish to place an "originator controlled" ("ORCON") restriction on a certain document, such that the document may be transmitted and used only by those specific other users whom he designates (and only in certain, expressly authorized ways). Such a restriction may be flexible if the "distribution list" could be modified after the creation of the document, specifically in the event of someone requesting permission from the originating user to transmit the document outside the original list of authorized recipients. The originating user may wish to permit distribution only to specific users, defined groups of users, defined geographic areas, users authorized to act in specific organizational roles, or a combination of any or all such attributes.

In this example, the organization may also desire to permit users to define a weaker distribution restriction such that access to a document is limited as above, but certain or all information within the document may be extracted and redistributed without  
5 further restriction by the recipients.

The organization and/or originating users may wish to know to what uses or geographic locations a document has been distributed. The organization may wish to know where  
10 documents with certain protection attributes have been distributed, for example, based on geographic information stored in site configuration records and/or name services records.

A user may wish to request a "return receipt" for a  
15 distributed document, or may wish to receive some indication of how a document has been handled by its recipients (e.g., whether it has been viewed, printed, edited and/or stored), for example, by specifying one or more audit requirements (or methods known to have audit requirements) in a PERC associated with such  
20 document(s).

### **User Environment**

In an organization (or association) such as that described above, users may utilize a variety of electronic appliances 600 for

processing and managing documents. This may include personal computers, both networked and otherwise, powerful single-user workstations, and servers or mainframe computers. To provide support for the control information described in this example, each electronic appliance that participates in use and management of VDE-protected documents may be enhanced with a VDE secure subsystem supporting an SPE 503 and/or HPE 655.

10           In some organizations, where the threats to secure operation are relatively low, an HPE 655 may suffice. In other organizations (e.g., government defense), it may be necessary to employ an SPE 503 in all situations where VDE-protected documents are processed. The choice of enhancement environment and technology may be different in different of the organization. Even if different types of PPE 650 are used within an organization to serve different requirements, they may be compatible and may operate on the same types (or subsets of types) of documents.

20

Users may employ application programs that are customized to operate in cooperation with the VDE for handling of VDE-protected documents. Examples of this may include VDE-aware document viewers, VDE aware electronic mail

systems, and similar applications. Those programs may communicate with the PPE 650 component of a user's electronic appliance 600 to make VDE-protected documents available for use while limiting the extent to which their contents may be  
5 copied, stored, viewed, modified, and/or transmitted and/or otherwise further distributed outside the specific electronic appliance.

Users may wish to employ commercial, off-the-shelf  
10 ("COTS") operating systems and application programs to process the VDE-protected documents. One approach to permit the use of COTS application programs and operating systems would be to allow such use only for documents without restrictions on redistribution. The standard VDE operating system redirector  
15 would allow users to access VDE-protected documents in a manner equivalent to that for files. In such an approach, however, a chain of control for metering and/or auditing use may be "broken" to some extent at the point that the protected object was made available to the COTS application. The fingerprinting  
20 (watermarking) techniques of VDE may be used to facilitate further tracking of any released information.

A variety of techniques may be used to protect printing of protected documents, such as, for example: server-based decryption engines, special fonts for "fingerprinting," etc.

5           Another approach to supporting COTS software would use the VDE software running on the user's electronic appliance to create one or more "virtual machine" environments in which COTS operating system and application programs may run, but from which no information may be permanently stored or  
10 otherwise transmitted except under control of VDE. Such an environment would permit VDE to manage all VDE-protected information, yet may permit unlimited use of COTS applications to process that information within the confines of a restricted environment. The entire contents of such an environment could  
15 be treated by VDE 100 as an extension to any VDE-protected documents read into the environment. Transmission of information out of the environment could be governed by the same rules as the original document(s).

## 20       **"Coarse-Grain" Control Capabilities**

As mentioned above, an organization may employ VDE-enforced control capabilities to manage the security, distribution, integrity, and control of entire documents. Some examples of these capabilities may include:

- 5                   1)     A communication channel connecting two or more electronic appliances 600 may be assigned a set of permitted sensitivity attributes. Only documents whose sensitivity attributes belong to this set would be permitted to be transmitted over the channel. This could be used to support the Device Labels requirement of the Trusted Computer System Evaluation Criteria (TCSEC).
- 10                   2)     A writable storage device (e.g., fixed disk, diskette, tape drive, optical disk) connected to or incorporated in an electronic appliance 600 may be assigned a set of permitted sensitivity attributes. Only documents whose sensitivity attributes belong to this set would be permitted to be stored on the device. This could be used to support the TCSEC Device Labels requirement.
- 15
- 20                   3)     A document may have a list of users associated with it representing the users who are permitted to "handle" the document. This list of users may represent, for example, the only users who may view the document, even if other users receive the document container, they could not manipulate the

contents. This could be used to support the standard ORCON handling caveat.

- 5 4) A document may have an attribute designating its originator and requiring an explicit permission to be granted by an originator before the document's content could be viewed. This request for permission may be made at the time the document is accessed by a user, or, for example, at the time one  
10 user distributes the document to another user. If permission is not granted, the document could not be manipulated or otherwise used.
- 15 5) A document may have an attribute requiring that each use of the document be reported to the document's originator. This may be used by an originator to gauge the distribution of the document. Optionally, the report may be required to have been  
20 made successfully before any use of the document is permitted, to ensure that the use is known to the controlling party at the time of use. Alternatively, for example, the report could be made in a deferred ("batch") fashion.



- 5                   6)    A document may have an attribute requiring that each use of the document be reported to a central document tracking clearinghouse. This could be used by the organization to track specific documents, to identify documents used by any particular user and/or group of users to track documents with specific attributes (e.g., sensitivity), etc. Optionally, for example, the report may be required to have been made successfully before any use of the document is permitted.
- 10
- 15                   7)    A VDE protected document may have an attribute requiring that each use of the document generate a "return receipt," to an originator. A person using the document may be required to answer specific questions in order to generate a return receipt, for example by indicating why the document is of interest, or by indicating some knowledge of the document's contents (after reading it). This may be used as assurance that the document had been handled by a person, not by any automated software mechanism.
- 20

- 5 8) A VDE protected document's content may be made available to a VDE-unaware application program in such a way that it is uniquely identifiable (traceable) to a user who caused its release. Thus, if the released form of the document is further distributed, its origin could be determined. This may be done by employing VDE "fingerprinting" for content release. Similarly, a printed VDE protected document may be marked in a similar, VDE
- 10 fingerprinted unique way such that the person who originally printed the document could be determined, even if copies have since been made.
- 15 9) Usage of VDE protected documents could be permitted under control of budgets that limit (based on size, time of access, etc.) access or other usage of document content. This may help prevent wholesale disclosure by limiting the number of VDE
- 20 documents accessible to an individual during a fixed time period. For example, one such control might permit a user, for some particular class of documents, to view at most 100 pages/day, but only print 10 pages/day and permit printing only on weekdays between nine and five. As a further

5 example, a user might be restricted to only a certain  
quantity of logically related, relatively "contiguous"  
and/or some other pattern (such as limiting the use  
of a database's records based upon the quantity of  
10 records that share a certain identifier in field) of  
VDE protected document usage to identify, for  
example, the occurrence of one or more types of  
excessive database usage (under normal or any  
reasonable circumstances). As a result, VDE  
15 content providers can restrict usage of VDE content  
to acceptable usage characteristics and thwart  
and/or identify (for example, by generating an  
exception report for a VDE administrator or  
organization supervisor) user attempts to  
20 inappropriately use, for example, such an  
information database resource.

These control capabilities show some examples of how  
VDE can be used to provide a flexible, interactive environment  
20 for tracking and managing sensitive documents. Such an  
environment could directly trace the flow of a document from  
person to person, by physical locations, by organizations, etc. It  
would also permit specific questions to be answered such as  
"what persons outside the R&D department have received any

R&D-controlled document." Because the control information is carried with each copy of a VDE protected document, and can ensure that central registries are updated and/or that originators are notified of document use, tracking can be prompt and accurate.

5

This contrasts with traditional means of tracking paper documents: typically, a paper-oriented system of manually collected and handled receipts is used. Documents may be individually copy-numbered and signed for, but once distributed are not actively controlled. In a traditional paper-oriented system, it is virtually impossible to determine the real locations of documents; what control can be asserted is possible only if all parties strictly follow the handling rules (which are at best inconvenient).

10

15

The situation is no better for processing documents within the context of ordinary computer and network systems.

Although said systems can enforce access control information based on user identity, and can provide auditing mechanisms for tracking accesses to files, these are low-level mechanisms that do not permit tracking or controlling the flow of content. In such systems, because document content can be freely copied and manipulated, it is not possible to determine where document

20

content has gone, or where it came from. In addition, because the control mechanisms in ordinary computer operating systems operate at a low level of abstraction, the entities they control are not necessarily the same as those that are manipulated by users. This particularly causes audit trails to be cluttered with voluminous information describing uninteresting activities.

### **Fine-Grain\* Control Capabilities**

In addition to controlling and managing entire documents, users may employ customized VDE-aware application software to control and manage individual modifications to documents. Examples of these capabilities include the following:

- 1) A VDE content user may be permitted to append further information to a VDE document to indicate a proposed alternative wording. This proposed alteration would be visible to all other users (in addition to the original text) of the document but would (for example) be able to be incorporated into the actual text only by the document's owner.
- 2) A group of VDE users could be permitted to modify one or more parts of a document in such a way that each individual alteration would be unambiguously

traceable to the specific user who performed it. The rights to modify certain portions of a document, and the extension of differing sets of rights to different users, allows an organization or secure environment to provide differing permissions enabling different rights to users of the same content.

- 3) A group of users could create a VDE document incrementally, by building it from individual contributions. These contributions would be bound together within a single controlled document, but each would be individually identified, for example, through their incorporation in VDE content containers as embedded container objects.

- 4) VDE control and management capabilities could be used to track activities related to individual document areas, for instance recording how many times each section of a document was viewed.

#### **Example - VDE Protected Content Repository**

As the "Digital Highway" emerges, there is increased discussion concerning the distribution of content across networks and, in particular, public networks such as the Internet. Content

may be made available across public networks in several ways including:

- 5                   •     "mailing" content to a user in response to a request or advance purchase (sending a token representing the commitment of electronic funds or credit to purchase an item);
  
- 10                  •     supporting content downloadable from an organization's own content repository, such a repository comprising, for example, a store of products (such as software programs) and/or a store of information resources, normally organized into one or more databases; and
  
- 15                  •     supporting a public repository into which other parties can deposit their products for redistribution to customers (normally by making electronic copies for distribution to a customer in response to a request).

20

One possible arrangement of VDE nodes involves use of one or more "repositories." A repository, for example, may serve as a location from which VDE participants may retrieve VDE content containers. In this case, VDE users may make use of a

network to gain access to a "server" system that allows one or more VDE users to access an object repository containing VDE content containers.

5           Some VDE participants may create or provide content and/or VDE content container objects, and then store content and/or content objects at a repository so that other participants may access such content from a known and/or efficiently organized (for retrieval) location. For example, a VDE repository  
10 (portion of a VDE repository, multiple VDE repositories, and/or providers of content to such repositories) may advertise the availability of certain types of VDE protected content by sending out email to a list of network users. If the network users have secure VDE subsystems in their electronic appliances, they may  
15 then choose to access such a repository directly, or through one or more smart agents and, using an application program for example, browse (and/or electronically search) through the offerings of VDE managed content available at the repository, download desirable VDE content containers, and make use of  
20 such containers. If the repository is successful in attracting users who have an interest in such content, VDE content providers may determine that such a repository is a desirable location(s) to make their content available for easy access by users. If a repository, such as CompuServe, stores content in



non-encrypted (plaintext) form, it may encrypt "outgoing" content on an "as needed" basis through placing such content in VDE content containers with desired control information, and may employ VDE secure communications techniques for content communication to VDE participants.

VDE repositories may also offer other VDE services. For example, a repository may choose to offer financial services in the form of credit from the repository that may be used to pay fees associated with use of VDE objects obtained from the repository. Alternatively or in addition, a VDE repository may perform audit information clearinghouse services on behalf of VDE creators or other participants (e.g. distributors, redistributors, client administrators, etc.) for usage information reported by VDE users. Such services may include analyzing such usage information, creating reports, collecting payments, etc.

A "full service" VDE repository may be very attractive to both providers and users of VDE managed content. Providers of VDE managed content may desire to place their content in a location that is well known to users, offers credit, and/or performs audit services for them. In this case, providers may be able to focus on creating content, rather than managing the

administrative processes associated with making content available in a "retail" fashion, collecting audit information from many VDE users, sending and receiving bills and payments, etc. VDE users may find the convenience of a single location (or an integrated arrangement of repositories) appealing as they are attempting to locate content of interest. In addition, a full service VDE repository may serve as a single location for the reporting of usage information generated as a consequence of their use of VDE managed content received from a VDE repository and/or, for example, receiving updated software (e.g. VDE-aware applications, load modules, component assemblies, non VDE-aware applications, etc.) VDE repository services may be employed in conjunction with VDE content delivery by broadcast and/or on physical media, such as CD-ROM, to constitute an integrated array of content resources that may be browsed, searched, and/or filtered, as appropriate, to fulfill the content needs of VDE users.

A public repository system may be established and maintained as a non-profit or for-profit service. An organization offering the service may charge a service fee, for example, on a per transaction basis and/or as a percentage of the payments by, and/or cost of, the content to users. A repository service may supply VDE authoring tools to content creators, publishers,

distributors, and/or value adding providers such that they may apply rules and controls that define some or all of the guidelines managing use of their content and so that they may place such content into VDE content container objects.

5

A repository may be maintained at one location or may be distributed across a variety of electronic appliances, such as a variety of servers (e.g. video servers, etc.) which may be at different locations but nonetheless constitute a single resource.

10

A VDE repository arrangement may employ VDE secure communications and VDE node secure subsystems ("protected processing environments"). The content comprising a given collection or unit of information desired by a user may be spread across a variety of physical locations. For example, content

15

representing a company's closing stock price and the activity (bids, lows, highs, etc.) for the stock might be located at a World Wide Web server in New York, and content representing an analysis of the company (such as a discussions of the company's history, personnel, products, markets, and/or competitors) might

20

be located on a server in Dallas. The content might be stored using VDE mechanisms to secure and audit use. The content might be maintained in clear form if sufficient other forms of security are available at such one or more of sites (e.g. physical security, password, protected operating system, data encryption,

or other techniques adequate for a certain content type). In the latter instances, content may be at least in part encrypted and placed in VDE containers as it streams out of a repository so as to enable secure communication and subsequent VDE usage control and usage consequence management.

A user might request information related to such a company including stock and other information. This request might, for example, be routed first through a directory or a more sophisticated database arrangement located in Boston. This arrangement might contain pointers to, and retrieve content from, both the New York and Dallas repositories. This information content may, for example, be routed directly to the user in two containers (e.g. such as a VDE content container object from Dallas and a VDE content container object from New York). These two containers may form two VDE objects within a single VDE container (which may contain two content objects containing the respective pieces of content from Dallas and New York) when processed by the user's electronic appliance.

Alternatively, such objects might be integrated together to form a single VDE container in Boston so that the information can be delivered to the user within a single container to simplify registration and control at the user's site. The information content from both locations may be stored as separate

information objects or they may be joined into a single, integrated information object (certain fields and/or categories in an information form or template may be filled in by one resource and other fields and/or categories may be filled by information provided by a different resource). A distributed database may manage such a distributed repository resource environment and use VDE to secure the storing, communicating, auditing, and/or use of information through VDE's electronic enforcement of VDE controls. VDE may then be used to provide both consistent content containers and content control services.

An example of one possible repository arrangement 3300 is shown in Figure 78. In this example, a repository 3302 is connected to a network 3304 that allows authors 3306A, 3306B, 3306C, and 3306D; a publisher 3308; and one or more end users 3310 to communicate with the repository 3302 and with each other. A second network 3312 allows the publisher 3308, authors 3306E and 3306F, an editor 3314, and a librarian 3316 to communicate with each other and with a local repository 3318. The publisher 3308 is also directly connected to author 3306E. In this example, the authors 3306 and publisher 3308 connect to the repository 3302 in order to place their content into an environment in which end users 3310 will be able to gain access to a broad selection of content from a common location.

In this example, the repository has two major functional areas: a content system 3302A and a clearinghouse system 3302B. The content system 3302A is comprised of a user/author registration system 3320, a content catalog 3322, a search mechanism 3324, content storage 3326, content references 3328, and a shipping system 3330 comprised of a controls packager 3322, a container packager 3334, and a transaction system 3336. The clearinghouse system 3302B is comprised of a user/author registration system 3338; template libraries 3340; a control structure library 3342; a disbursement system 3344; an authorization system 3346 comprised of a financial system 3348 and a content system 3350; a billing system 3352 comprised of a paper system 3354, a credit card system 3356, and an electronic funds transfer (EFT) system 3358; and an audit system 3360 comprised of a receipt system 3362, a response system 3364, a transaction system 3366, and an analysis system 3368.

In this example, author 3306A creates content in electronic form that she intends to make broadly available to many end users 3310, and to protect her rights through use of VDE. Author 3306A transmits a message to the repository 3302 indicating her desire to register with the repository to distribute her content. In response to this message, the user/author registration system 3320 of the content system 3302A, and the

user/author registration system 3338 of the clearinghouse system 3302B transmit requests for registration information to author 3306A using the network 3304. These requests may be made in an on-line interactive mode; or they may be transmitted in a batch to author 3306A who then completes the requested information and transmits it as a batch to the repository 3302; or some aspects may be handled on-line (such as basic identifying information) and other information may be exchanged in a batch mode.

10

Registration information related to the content system 3302A may, for example, include:

- a request that Author 3306A provide information concerning the types and/or categories of content proposed for storage and access using the repository,
- the form of abstract and/or other identifying information required by the repository—in addition to providing author 3306A with an opportunity to indicate whether or not author 3306A generally includes other information with content submissions (such as promotional materials, detailed information regarding the format of submitted content, any equipment requirements that should or must be met

15

20

for potential users of submitted content to  
successfully exploit its value, etc.),

- 5                   ●     requests for information from author 3306A  
concerning where the content is to be located (stored  
at the repository, stored at author 3306A's location,  
stored elsewhere, or some combination of locations),
- 10                  ●     what general search characteristics should be  
associated with content submissions (e.g. whether  
abstracts should be automatically indexed for  
searches by users of the repository, the manner in  
which content titles, abstracts, promotional  
materials, relevant dates, names of performers  
15                  and/or authors, or other information related to  
content submissions may or should be used in lists  
of types of content and/or in response to searches,  
etc.), and/or
- 20                  ●     how content that is stored at and/or passed through  
the repository should be shipped (including any  
container criteria, encryption requirements,  
transaction requirements related to content  
transmissions, other control criteria, etc.)



The information requested from author 3306A by the user/author registration system of the clearinghouse may, for example, consist of:

- 5                   •     VDE templates that author 3306A may or must make use of in order to correctly format control information such that, for example, the audit system 3360 of the clearinghouse system 3302B is properly authorized to receive and/or process usage information related to content submitted by author 10                   3306A,
  
- VDE control information available from the clearinghouse 3302B that may or must be used by author 3306A (and/or included by reference) in some 15                   or all of the VDE component assemblies created and/or used by author 3306A associated with submitted content,
  
- the manner in which disbursement of any funds 20                   associated with usage of content provided by, passed through, or collected by the repository clearinghouse system 3302B should be made,

- the form and/or criteria of authorizations to use submitted content and/or financial transactions associated with content,
- 5 • the acceptable forms of billing for use of content and/or information associated with content (such as analysis reports that may be used by others),
- 10 • how VDE generated audit information should be received,
- how responses to requests from users should be managed,
- 15 • how transactions associated with the receipt of audit information should be formatted and authorized,
- how and what forms of analysis should be performed on usage information, and/or
- 20 • under what circumstances (if any) usage information and/or analysis results derived from VDE controlled content usage information should be managed (including to whom they may or must be delivered,

the form of delivery, any control information that may be associated with use of such information, etc.)

5       The repository 3302 receives the completed registration  
information from author 3306A and uses this information to  
build an account profile for author 3306A. In addition, software  
associated with the authoring process may be transmitted to  
author 3306A. This software may, for example, allow author  
3306A to place content into a VDE content container with  
10       appropriate controls in such a way that many of the decisions  
associated with creating such containers are made automatically  
to reflect the use of the repository 3302 as a content system  
and/or a clearinghouse system (for example, the location of  
content, the party to contact for updates to content and/or  
15       controls associated with content, the party or parties to whom  
audit information may and/or must be transmitted and the  
pathways for such communication, the character of audit  
information that is collected during usage, the forms of payment  
that are acceptable for use of content, the frequency of audit  
20       transmissions required, the frequency of billing, the form of  
abstract and/or other identifying information associated with  
content, the nature of at least a portion of content usage control  
information, etc.)

Author 3306A makes use of a VDE authoring application to specify the controls and the content that she desires to place within a VDE content container, and produces such a container in accordance with any requirements of the repository 3302.

5       Such a VDE authoring application may be, for example, an application provided by the repository 3302 which can help ensure adherence to repository content control requirements such as the inclusion of one or more types of component assemblies or other VDE control structures and/or required  
10       parameter data, an application received from another party, and/or an application created by author 3306A in whole or in part. Author 3306A then uses the network 3304 to transmit the container and any deviations from author 3306A's account profile that may relate to such content to the repository 3302. The  
15       repository 3302 receives the submitted content, and then -- in accordance with any account profile requirements, deviations and/or desired options in this example—makes a determination as to whether the content was produced within the boundaries of any content and/or control information requirements of the  
20       repository and therefore should be placed within content storage or referenced by a location pointer or the like. In addition to placing the submitted content into content storage or referencing such content's location, the repository 3302 may also make note of characteristics associated with such submitted content in the

search mechanism 3324, content references 3328, the shipping  
system 3330, and/or the relevant systems of the clearinghouse  
system 3302B related to templates and control structures,  
authorizations, billing and/or payments, disbursements, and/or  
5 audits of usage information.

During an authoring process, author 3306A may make use  
of VDE templates. Such templates may be used as an aspect of a  
VDE authoring application. For example, such templates may be  
10 used in the construction of a container as described above.  
Alternatively or in addition, such templates may also be used  
when submitted content is received by the repository 3302.  
References to such templates may be incorporated by author  
3306A as an aspect of constructing a container for submitted  
15 content (in this sense the container delivered to the repository  
may be in some respects "incomplete" until the repository  
"completes" the container through use of indicated templates).  
Such references may be required for use by the repository 3302  
(for example, to place VDE control information in place to fulfill  
20 an aspect of the repository's business or security models such as  
one or more map tables corresponding to elements of content  
necessary for interacting with other VDE control structures to  
accommodate certain metering, billing, budgeting, and/or other  
usage and/or distribution related controls of the repository).

For example, if content submitted by author 3306A consists of a periodical publication, a template delivered to the author by the repository 3302 when the author registers at the repository may be used as an aspect of an authoring application manipulated by the author in creating a VDE content container for such a periodical. Alternatively or in addition, a template designed for use with periodical publications may be resident at the repository 3302, and such a template may be used by the repository to define, in whole or in part, control structures associated with such a container. For example, a VDE template designed to assist in formulating control structures for periodical publications might indicate (among other things) that:

- usage controls should include a meter method that records each article within a publication that a user opens,
- a certain flat rate fee should apply to opening the periodical regardless of the number of articles opened, and/or
- a record should be maintained of every advertisement that is viewed by a user.

If content is maintained in a known and/or identifiable format, such a template may be used during initial construction of a container without author 3306A's intervention to identify any map tables that may be required to support such recording and billing actions. If such a VDE template is unavailable to author 3306A, she may choose to indicate that the container submitted should be reconstructed (e.g. augmented) by the repository to include the VDE control information specified in a certain template or class of templates. If the format of the content is known and/or identifiable by the repository, the repository may be able to reconstruct (or "complete") such a container automatically.

One factor in a potentially ongoing financial relationship between the repository and author 3306A may relate to usage of submitted content by end users 3310. For example, author 3306A may negotiate an arrangement with the repository wherein the repository is authorized to keep 20% of the total revenues generated from end users 3310 in exchange for maintaining the repository services (e.g. making content available to end users 3310, providing electronic credit, performing billing activities, collecting fees, etc.) A financial relationship may be recorded in control structures in flexible and configurable ways. For example, the financial relationship

described above could be created in a VDE container and/or installation control structure devised by author 3306A to reflect author 3306A's financial requirements and the need for a 20% split in revenue with the repository wherein all billing activities related to usage of submitted content could be processed by the repository, and control structures representing reciprocal methods associated with various component assemblies required for use of author 3306A's submitted content could be used to calculate the 20% of revenues. Alternatively, the repository may independently and securely add and/or modify control structures originating from author 3306A in order to reflect an increase in price. Under some circumstances, author 3306A may not be directly involved (or have any knowledge of) the actual price that the repository charges for usage activities, and may concern herself only with the amount of revenue and character of usage analysis information that she requires for her own purposes, which she specifies in VDE control information which governs the use, and consequences of use, of VDE controlled content.

Another aspect of the relationship between authors and the repository may involve the character of transaction recording requirements associated with delivery of VDE controlled content and receipt of VDE controlled content usage audit information. For example, author 3306A may require that the repository



make a record of each user that receives a copy of content from the repository. Author 3306A may further require collection of information regarding the circumstances of delivery of content to such users (e.g. time, date, etc.) In addition, the repository may  
5 elect to perform such transactions for use internally (e.g. to determine patterns of usage to optimize systems, detect fraud, etc.)

In addition to recording information regarding delivery of  
10 such VDE controlled content, author 3306A may have required or requested the repository to perform certain VDE container related processes. For example, author 3306A may want differing abstract and/or other descriptive information delivered to different classes of users. In addition, author 3306A may wish  
15 to deliver promotional materials in the same container as submitted content depending on, for example, the character of usage exhibited by a particular user (e.g. whether the user has ever received content from author 3306A, whether the user is a regular subscriber to author 3306A's materials, and/or other  
20 patterns that may be relevant to author 3306A and/or the end user that are used to help determine the mix of promotional materials delivered to a certain VDE content end user.) In another example, author 3306A may require that VDE

fingerprinting be performed on such content prior to transmission of content to an end user.

5 In addition to the form and/or character of content shipped to an end user, authors may also require certain encryption related processes to be performed by the repository as an aspect of delivering content. For example, author 3306A may have required that the repository encrypt each copy of shipped content using a different encryption key or keys in order to help  
10 maintain greater protection for content (e.g. in case an encryption key was "cracked" or inadvertently disclosed, the "damage" could be limited to the portion(s) of that specific copy of a certain content deliverable). In another example, encryption functions may include the need to use entirely different  
15 encryption algorithms and/or techniques in order to fulfill circumstantial requirements (e.g. to comply with export restrictions). In a further example, encryption related processes may include changing the encryption techniques and/or algorithms based on the level of trustedness and/or tamper  
20 resistance of the VDE site to which content is delivered.

In addition to transaction information gathered when content is shipped from a VDE repository to an end user, the repository may be required to keep transaction information

related to the receipt of usage information, requests, and/or responses to and/or from end users 3310. For example, author 3306A may require the repository to keep a log of some or all connections made by end users 3310 related to transmissions and or reception of information related to the use of author 3306A's content (e.g. end user reporting of audit information, end user requests for additional permissions information, etc.)

Some VDE managed content provided to end users 3310 through the repository may be stored in content storage. Other information may be stored elsewhere, and be referenced through the content references. In the case where content references are used, the repository may manage the user interactions in such a manner that all repository content, whether stored in content storage or elsewhere (such as at another site), is presented for selection by end users 3310 in a uniform way, such as, for example, a consistent or the same user interface. If an end user requests delivery of content that is not stored in content storage, the VDE repository may locate the actual storage site for the content using information stored in content references (e.g. the network address where the content may be located, a URL, a filesystem reference, etc.) After the content is located, the content may be transmitted across the network to the repository or it may be delivered directly from where it is stored to the

5

10

15

20

charge for certain content usage activities, differences in charges based on whether or not certain usage information is recorded and/or made available to the repository and/or content provider, the redistribution rights associated with such content, the  
5 reporting frequency for audit transmissions, the forms of credit and/or currency that may be used to pay certain fees associated with use of such content, discounts related to certain volumes of usage, discounts available due to the presence of rights associated with other content from the same and/or different  
10 content providers, sales, etc. Furthermore, a VDE repository content catalog 3322 may indicate some or all of the component assemblies that are required in order to make use of content such that the end user's system and the repository can exchange messages to help ensure that any necessary VDE component  
15 assemblies or other VDE control information is identified, and if necessary and authorized, are delivered along with such content to the end user (rather than, for example, being requested later after their absence has been detected during a registration and/or use attempt).

20

In order to make use of the VDE repository in this example, an end user must register with the repository. In a manner similar to that indicated above in the case of an author, a VDE end user transmits a message from her VDE installation

to the repository across the network indicating that she wishes to make use of the services provided by the repository (e.g. access content stored at and/or referenced by the repository, use credit provided by the repository, etc.) In response to this message, the

5 user/author registration systems of the content system 3302A and the clearinghouse system 3302B of the repository transmit requests for information from the end user (e.g. in an on-line and/or batch interaction). The information requested by the user/author registration system of the content system 3302A may

10 include type(s) of content that the user wishes to access, the characteristics of the user's electronic appliance 600, etc. The information requested by the user/author registration system of the clearinghouse system 3302B may include whether the user wishes to establish a credit account with the clearinghouse

15 system 3302B, what other forms of credit the user may wish to use for billing purposes, what other clearinghouses may be used by the end user in the course of interacting with content obtained from the repository, any general rules that the user has established regarding their preferences for release and handling

20 of usage analysis information, etc. Once the end user has completed the registration information and transmitted it to the repository, the repository may construct an account profile for the user. In this example, such requests and responses are

handled by secure VDE communications between secure VDE subsystems of both sending and receiving parties.

5           In order to make use of the repository, the end user may operate application software. In this example, the end user may either make use of a standard application program (e.g. a World Wide Web browser such as Mosaic), or they may make use of application software provided by the repository after completion of the registration process. If the end user chooses to make use of the application software provided by the repository, they may be able to avoid certain complexities of interaction that may occur if a standard package is used. Although standardized packages are often relatively easy to use, a customized package that incorporates VDE aware functionality may provide an easier to use interface for a user. In addition, certain characteristics of the repository may be built in to the interface to simplify use of the services (e.g. similar to the application programs provided by America Online).

20           The end user may connect to the repository using the network. In this example, after the user connects to the repository, an authentication process will occur. This process can either be directed by the user (e.g. through use of a login and password protocol) or may be established by the end user's

electronic appliance secure subsystems interacting with a repository electronic appliance in a VDE authentication. In either event, the repository and the user must initially ensure that they are connected to the correct other party. In this  
5 example, if secured information will flow between the parties, a VDE secured authentication must occur, and a secure session must be established. On the other hand, if the information to be exchanged has already been secured and/or is available without authentication (e.g. certain catalog information, containers that  
10 have already been encrypted and do not require special handling, etc.), the "weaker" form of login/password may be used.

Once an end user has connected to the VDE repository and authentication has occurred, the user may begin manipulating  
15 and directing their user interface software to browse through a repository content catalog 3322 (e.g. lists of publications, software, games, movies, etc.), use the search mechanism to help locate content of interest, schedule content for delivery, make inquiries of account status, availability of usage analysis  
20 information, billing information, registration and account profile information, etc. If a user is connecting to obtain content, the usage requirements for that content may be delivered to them. If the user is connecting to deliver usage information to the repository, information related to that transmission may be



delivered to them. Some of these processes are described in more detail below.

5 In this example, when an end user requests content from the VDE repository (e.g. by selecting from a menu of available options), the content system 3302A locates the content either in the content references and/or in content storage. The content system 3302A may then refer to information stored in the content catalog 3322, the end user's account profile, and/or the author's account profile to determine the precise nature of  
10 container format and/or control information that may be required to create a VDE content container to fulfill the end user's request. The shipping system then accesses the clearinghouse system 3302B to gather any necessary additional control  
15 structures to include with the container, to determine any characteristics of the author's and/or end user's account profiles that may influence either the transaction(s) associated with delivering the content to the end user or with whether the transaction may be processed. If the transaction is authorized,  
20 and all elements necessary for the container are available, the controls packager forms a package of control information appropriate for this request by this end user, and the container packager takes this package of control information and the content and forms an appropriate container (including any

permissions that may be codeliverable with the container, incorporating any encryption requirements, etc.) If required by the repository or the author's account profile, transactions related to delivery of content are recorded by the transaction system of the shipping system. When the container and any transactions related to delivery have been completed, the container is transmitted across the network to the end user.

An end user may make use of credit and/or currency securely stored within the end user's VDE installation secure subsystem to pay for charges related to use of VDE content received from the repository, and/or the user may maintain a secure credit and/or currency account remotely at the repository, including a "virtual" repository where payment is made for the receipt of such content by an end user. This later approach may provide greater assurance for payment to the repository and/or content providers particularly if the end user has only an HPE based secure subsystem. If an end user electronic credit and/or currency account is maintained at the repository in this example, charges are made to said account based on end user receipt of content from the repository. Further charges to such a remote end user account may be made based on end user usage of such received content and based upon content usage information communicated to the repository clearinghouse system 3302B.

In this example, if an end user does not have a relationship established with a financial provider (who has authorized the content providers whose content may be obtained through use of the repository to make use of their currency and/or credit to pay for any usage fees associated with such provider's content) and/or if an end user desires a new source of such credit, the end user may request credit from the repository clearinghouse system 3302B. If an end user is approved for credit, the repository may extend credit in the form of credit amounts (e.g. recorded in one or more UDEs) associated with a budget method managed by the repository. Periodically, usage information associated with such a budget method is transmitted by the end user to the audit system of the repository. After such a transmission (but potentially before the connection is terminated), an amount owing is recorded for processing by the billing system, and in accordance with the repository's business practices, the amount of credit available for use by the end user may be replenished in the same or subsequent transmission. In this example, the clearinghouse of the repository supports a billing system with a paper system for resolving amounts owed through the mail, a credit card system for resolving amounts owed through charges to one or more credit cards, and an electronic funds transfer system for resolving such amounts through direct debits to a bank account. The repository may

automatically make payments determined by the disbursement system for monies owed to authors through use of similar means. Additional detail regarding the audit process is provided below.

5           As indicated above, end users 3310 in this example will periodically contact the VDE repository to transmit content usage information (e.g. related to consumption of budget, recording of other usage activities, etc.), replenish their budgets, modify their account profile, access usage analysis information, and perform other administrative and information exchange activities. In some cases, an end user may wish to contact the repository to obtain additional control structures. For example, if an end user has requested and obtained a VDE content container from the repository, that container is typically shipped to the end user along with control structures appropriate to the content, the author's requirements and account profile, the end user's account profile, the content catalog 3322, and/or the circumstances of the delivery (e.g. the first delivery from a particular author, a subscription, a marketing promotion, presence and/or absence of certain advertising materials, requests formulated on behalf of the user by the user's local VDE instance, etc.) Even though, in this example, the repository may have attempted to deliver all relevant control structures, some containers may include controls structures that allow for options

that the end user did not anticipate exercising (and the other criteria did not automatically select for inclusion in the container) that the end user nonetheless determines that they would like to exercise. In this case, the end user may wish to  
5 contact the repository and request any additional control information (including, for example, control structures) that they will need in order to make use of the content under such option.

For example, if an end user has obtained a VDE content  
10 container with an overall control structure that includes an option that records of the number of times that certain types of accesses are made to the container and further bases usage fees on the number of such accesses, and another option within the overall control structure allows the end user to base the fees paid  
15 for access to a particular container based on the length of time spent using the content of the container, and the end user did not originally receive controls that would support this latter form of usage, the repository may deliver such controls at a later time and when requested by the user. In another example, an author  
20 may have made changes to their control structures (e.g. to reflect a sale, a new discounting model, a modified business strategy, etc.) which a user may or must receive in order to use the content container with the changed control structures. For example, one or more control structures associated with a certain VDE content

container may require a "refresh" for continued authorization to employ such structures, or the control structures may expire. This allows (if desired) a VDE content provider to periodically modify and/or add to VDE control information at an end user's site (employing the local VDE secure subsystem).

Audit information (related to usage of content received from the repository) in this example is securely received from end users 3310 by the receipt system 3362 of the clearinghouse. As indicated above, this system may process the audit information and pass some or all of the output of such a process to the billing system and/or transmit such output to appropriate content authors. Such passing of audit information employs secure VDE pathway of reporting information handling techniques. Audit information may also be passed to the analysis system in order to produce analysis results related to end user content usage for use by the end user, the repository, third party market researchers, and/or one or more authors. Analysis results may be based on a single audit transmission, a portion of an audit transmission, a collection of audit transmissions from a single end user and/or multiple end users 3310, or some combination of audit transmissions based on the subject of analysis (e.g. usage patterns for a given content element or collection of elements, usage of certain categories of

content, payment histories, demographic usage patterns, etc.)

5 The response system 3364 is used to send information to the end user to, for example, replenish a budget, deliver usage controls, update permissions information, and to transmit certain other information and/or messages requested and/or required by an end user in the course of their interaction with the clearinghouse. During the course of an end user's connections and transmissions to and from the clearinghouse, certain transactions (e.g. time, date, and/or purpose of a connection and/or transmission) may be recorded by the transaction system  
10 of the audit system to reflect requirements of the repository and/or authors.

Certain audit information may be transmitted to authors.  
15 For example, author 3306A may require that certain information gathered from an end user be transmitted to author 3306A with ~~no~~ processing by the audit system. In this case, the fact of the transmission may be recorded by the audit system, but author 3306A may have elected to perform their own usage analysis  
20 rather than (or in addition to) permitting the repository to access, otherwise process and/or otherwise use this information. The repository in this example may provide author 3306A with some of the usage information related to the repository's budget method received from one or more end users 3310 and generated

by the payment of fees associated with such users' usage of content provided by author 3306A . In this case, author 3306A may be able to compare certain usage information related to content with the usage information related to the repository's budget method for the content to analyze patterns of usage (e.g. to analyze usage in light of fees, detect possible fraud, generate user profile information, etc.) Any usage fees collected by the clearinghouse associated with author 3306A's content that are due to author 3306A will be determined by the disbursement system of the clearinghouse. The disbursement system may include usage information (in complete or summary form) with any payments to author 3306A resulting from such a determination. Such payments and information reporting may be an entirely automated sequence of processes occurring within the VDE pathway from end user VDE secure subsystems, to the clearinghouse secure subsystem, to the author's secure subsystem.

In this example, end users 3310 may transmit VDE permissions and/or other control information to the repository 3302 permitting and/or denying access to usage information collected by the audit system for use by the analysis system. This, in part, may help ensure end user's privacy rights as it relates to the usage of such information. Some containers may



require, as an aspect of their control structures, that an end user make usage information available for analysis purposes. Other containers may give an end user the option of either allowing the usage information to be used for analysis, or denying some or all such uses of such information. Some users may elect to allow analysis of certain information, and deny this permission for other information. End users 3310 in this example may, for example, elect to limit the granularity of information that may be used for analysis purposes (e.g. an end user may allow analysis of the number of movies viewed in a time period but disallow use of specific titles, an end user may allow release of their ZIP code for demographic analysis, but disallow use of their name and address, etc.) Authors and/or the repository 3302 may, for example, choose to charge end users 3310 smaller fees if they agree to release certain usage information for analysis purposes.

In this example, the repository 3302 may receive content produced by more than one author. For example, author B, author C, and author D may each create portions of content that will be delivered to end users 3310 in a single container. For example, author B may produce a reference work. Author C may produce a commentary on author B's reference work, and author D may produce a set of illustrations for author B's reference work and author C's commentary. Author B may collect together

author C's and author D's content and add further content (e.g. the reference work described above) and include such content in a single container which is then transmitted to the repository 3302. Alternatively, each of the authors may transmit their works to the repository 3302 independently, with an indication that a template should be used to combine their respective works prior to shipping a container to an end user. Still alternatively, a container reflecting the overall content structure may be transmitted to the repository 3302 and some or all of the content may be referenced in the content references rather than delivered to the repository 3302 for storage in content storage.

When an end user makes use of container content, their content usage information may, for example, be segregated in accordance with control structures that organize usage information based at least in part on the author who created that segment. Alternatively, the authors and/or the VDE repository 3302 may negotiate one or more other techniques for securely dividing and/or sharing usage information in accordance with VDE control information. Furthermore, control structures associated with a container may implement models that differentiate any usage fees associated with portions of content based on usage of particular portions, overall usage of the container, particular patterns of usage, or other mechanism

negotiated (or otherwise agreed to) by the authors. Reports of usage information, analysis results, disbursements, and other clearinghouse processes may also be generated in a manner that reflects agreements reached by repository 3302 participants (authors, end users 3310 and/or the repository 3302) with respect to such processes. These agreements may be the result of a VDE control information negotiation amongst these participants.

In this example, one type of author is a publisher 3308. The publisher 3308 in this example communicates over an "internal" network with a VDE based local repository 3302 and over the network described above with the public repository 3302. The publisher 3308 may create or otherwise provide content and/or VDE control structure templates that are delivered to the local repository 3302 for use by other participants who have access to the "internal" network. These templates may be used to describe the structure of containers, and may further describe whom in the publisher 3308's organization may take which actions with respect to the content created within the organization related to publication for delivery to (and/or referencing by) the repository 3302. For example, the publisher 3308 may decide (and control by use of said temple) that a periodical publication will have a certain format with respect to the structure of its content and the types

of information that may be included (e.g. text, graphics, multimedia presentations, advertisements, etc.), the relative location and/or order of presentation of its content, the length of certain segments, etc. Furthermore, the publisher 3308 may, for example, determine (through distribution of appropriate permissions) that the publication editor is the only party that may grant permissions to write into the container, and that the organization librarian is the only party that may index and/or abstract the content. In addition, the publisher 3308 may, for example, allow only certain one or more parties to finalize a container for delivery to the repository 3302 in usable form (e.g. by maintaining control over the type of permissions, including distribution permissions, that may be required by the repository 3302 to perform subsequent distribution activities related to repository end users 3310).

In this example, author 3306E is connected directly to the publisher 3308, such that the publisher 3308 can provide templates for that author that establish the character of containers for author 3306E's content. For example, if author 3306E creates books for distribution by the publisher 3308, the publisher 3308 may define the VDE control structure template which provides control method options for author 3306E to select from and which provides VDE control structures for securely

distributing author 3306E's works. Author 3306E and the publisher 3308 may employ VDE negotiations for the template characteristics, specific control structures, and/or parameter data used by author 3306E. Author 3306E may then use the  
5 template(s) to create control structures for their content containers. The publisher 3308 may then deliver these works to the repository 3302 under a VDE extended agreement comprising electronic agreements between author 3306E and the publisher 3308 and the repository 3302 and the publisher 3308.

10

In this example, the publisher 3308 may also make author 3306E's work available on the local repository 3302. The editor may authorize (e.g. through distribution of appropriate permissions) author F to create certain portions of content for a  
15 publication. In this example, the editor may review and/or modify author F's work and further include it in a container with content provided by author 3306E (available on the local repository 3302). The editor may or may not have permissions from the publisher 3308 to modify author 3306E's content  
20 (depending on any negotiation(s) that may have occurred between the publisher 3308 and author 3306E, and the publisher 3308's decision to extend such rights to the editor if permissions to modify author 3306E's content are held in redistributable form by the publisher 3308). The editor may also include content from

other authors by (a) using a process of granting permissions to authors to write directly into the containers and/or (b) retrieving containers from the local repository 3302 for inclusion. The local repository 3302 may also be used for other material used by the publisher 3308's organization (e.g. databases, other reference works, internal documents, draft works for review, training videos, etc.), such material may, given appropriate permissions, be employed in VDE container collections of content created by the editor.

10

The librarian in this example has responsibility for building and/or editing inverted indexes, keyword lists (e.g. from a restricted vocabulary), abstracts of content, revision histories, etc. The publisher 3308 may, for example, grant permissions to only the librarian for creating this type of content. The publisher 3308 may further require that this building and/or editing occur prior to release of content to the repository 3302.

15

#### **Example -- Evolution and Transformation of VDE Managed Content and Control Information**

20

The VDE content control architecture allows content control information (such as control information for governing content usage) to be shaped to conform to VDE control information requirements of multiple parties. Formulating such

25

multiple party content control information normally involves securely deriving control information from control information securely contributed by parties who play a role in a content handling and control model (e.g. content creator(s), provider(s), user(s), clearinghouse(s), etc.). Multiple party control information may be necessary in order to combine multiple pieces of independently managed VDE content into a single VDE container object (particularly if such independently managed content pieces have differing, for example conflicting, content control information). Such secure combination of VDE managed pieces of content will frequently require VDE's ability to securely derive content control information which accommodates the control information requirements, including any combinatorial rules, of the respective VDE managed pieces of content and reflects an acceptable agreement between such plural control information sets.

The combination of VDE managed content pieces may result in a VDE managed composite of content. Combining VDE managed content must be carried out in accordance with relevant content control information associated with said content pieces and processed through the use of one or more secure VDE sub-system PPEs 650. VDE's ability to support the embedding, or otherwise combining, of VDE managed content pieces, so as to

create a combination product comprised of various pieces of VDE content, enables VDE content providers to optimize their VDE electronic content products. The combining of VDE managed content pieces may result in a VDE content container which

5 "holds" consolidated content and/or concomitant, separate, nested VDE content containers.

VDE's support for creation of content containers holding distinct pieces of VDE content portions that were previously

10 managed separately allows VDE content providers to develop products whose content control information reflects value propositions consistent with the objectives of the providers of content pieces, and further are consistent with the objectives of a content aggregator who may be producing a certain content

15 combination as a product for commercial distribution. For example, a content product "launched" by a certain content provider into a commercial channel (such as a network repository) may be incorporated by different content providers and/or end-users into VDE content containers (so long as such

20 incorporation is allowed by the launched product's content control information). These different content providers and/or end-users may, for example, submit differing control information for regulating use of such content. They may also combine in different combinations a certain portion of launched content with



content received from other parties (and/or produced by themselves) to produce different content collections, given appropriate authorizations.

5           VDE thus enables copies of a given piece of VDE managed content to be securely combined into differing consolidations of content, each of which reflects a product strategy of a different VDE content aggregator. VDE's content aggregation capability will result in a wider range of competitive electronic content  
10 products which offer differing overall collections of content and may employ differing content control information for content that may be common to such multiple products. Importantly, VDE securely and flexibly supports editing the content in, extracting content from, embedding content into, and otherwise shaping the  
15 content composition of, VDE content containers. Such capabilities allow VDE supported product models to evolve by progressively reflecting the requirements of "next" participants in an electronic commercial model. As a result, a given piece of VDE managed content, as it moves through pathways of  
20 handling and branching, can participate in many different content container and content control information commercial models.

7/10/07/15243

VDE content, and the electronic agreements associated with said content, can be employed and progressively manipulated in commercial ways which reflect traditional business practices for non-electronic products (though VDE supports greater flexibility and efficiency compared with most of such traditional models). Limited only by the VDE control information employed by content creators, other providers, and other pathway of handling and control participants. VDE allows a "natural" and unhindered flow of, and creation of, electronic content product models. VDE provides for this flow of VDE products and services through a network of creators, providers, and users who successively and securely shape and reshape product composition through content combining, extracting, and editing within a Virtual Distribution Environment.

15

VDE provides means to securely combine content provided at different times, by differing sources, and or representing differing content types. These types, timings, and/or different sources of content can be employed to form a complex array of content within a VDE content container. For example, a VDE content container may contain a plurality of different content container objects, each containing different content whose usage can be controlled, at least in part, by its own container's set of VDE content control information.

5 A VDE content container object may, through the use of a secure VDE sub-system, be "safely" embedded within a "parent" VDE content container. This embedding process may involve the creation of an embedded object, or, alternatively, the containing, within a VDE content container, of a previously independent and now embedded object by, at minimum, appropriately referencing said object as to its location.

10 An embedded content object within a parent VDE content container:

15 (1) may have been a previously created VDE content container which has been embedded into a parent VDE content container by securely transforming it from an independent to an embedded object through the secure processing of one or more VDE component assemblies within a VDE secure sub-system PPE 650. In this instance, an embedded object may be subject to content control information, including one or more permissions records associated with the parent container, but may not, for example, have its own content control information other than content identification information, or the embedded object may be more extensively controlled by its own content control information (e.g. permissions records).

20

(2) may include content which was extracted from another VDE content container (along with content control information, as may be applicable) for inclusion into a parent VDE content container in the form of an embedded VDE content container object. In this case, said extraction and embedding may use one or more VDE processes which run securely within a VDE secure sub-system PPE 650 and which may securely remove (or copy) the desired content from a source VDE content container and place such content in a new or existing container object, either of which may be or become embedded into a parent VDE content container.

(3) may include content which was first created and then placed in a VDE content container object. Said receiving container may already be embedded in a parent VDE content container and may already contain other content. The container in which such content is placed may be specified using a VDE aware application which interacts with content and a secure VDE subsystem to securely create such VDE container and place such content therein followed by securely embedding such container into the destination, parent container. Alternatively, content may be specified without the use of a VDE aware

application, and then manipulated using a VDE aware application in order to manage movement of the content into a VDE content container. Such an application may be a VDE aware word processor, desktop and/or multimedia publishing package, graphics and/or presentation package, etc. It may also be an operating system function (e.g. part of a VDE aware operating system or mini-application operating with an O/S such as a Microsoft Windows compatible object packaging application) and movement of content from "outside" VDE to within a VDE object may, for example, be based on a "drag and drop" metaphor that involves "dragging" a file to a VDE container object using a pointing device such as a mouse. Alternatively, a user may "cut" a portion of content and "paste" such a portion into a VDE container by first placing content into a "clipboard," then selecting a target content object and pasting the content into such an object. Such processes may, at the direction of VDE content control information and under the control of a VDE secure subsystem, put the content automatically at some position in the target object, such as at the end of the object or in a portion of the object that corresponds to an identifier carried by or with the content such as a field identifier, or the embedding process might pop-up a user interface that allows a user to browse

a target object's contents and/or table of contents and/or other directories, indexes, etc. Such processes may further allow a user to make certain decisions concerning VDE content control information (budgets limiting use, reporting pathway(s), usage registration requirements, etc.) to be applied to such embedded content and/or may involve selecting the specific location for embedding the content, all such processes to be performed as transparently as practical for the application.

(4) may be accessed in conjunction with one or more operating system utilities for object embedding and linking, such as utilities conforming to the Microsoft OLE standard. In this case, a VDE container may be associated with an OLE "link." Accesses (including reading content from, and writing content to) to a VDE protected container may be passed from an OLE aware application to a VDE aware OLE application that accesses protected content in conjunction with control information associated with such content.

A VDE aware application may also interact with component assemblies within a PPE to allow direct editing of the content of a VDE container, whether the content is in a parent or

embedded VDE content container. This may include the use of a VDE aware word processor, for example, to directly edit (add to, delete, or otherwise modify) a VDE container's content. The secure VDE processes underlying VDE container content editing  
5 may be largely or entirely transparent to the editor (user) and may transparently enable the editor to securely browse through (using a VDE aware application) some or all of the contents of, and securely modify one or more of the VDE content containers embedded in, a VDE content container hierarchy.

10

The embedding processes for all VDE embedded content containers normally involves securely identifying the appropriate content control information for the embedded content. For example, VDE content control information for a VDE installation  
15 and/or a VDE content container may securely, and transparently to an embedder (user), apply the same content control information to edited (such as modified or additional) container content as is applied to one or more portions (including all, for example) of previously "in place" content of said container and/or  
20 securely apply control information generated through a VDE control information negotiation between control sets, and/or it may apply control information previously applied to said content. Application of control information may occur regardless of whether the edited content is in a parent or embedded container.

This same capability of securely applying content control information (which may be automatically and/or transparently applied), may also be employed with content that is embedded into a VDE container through extracting and embedding content, or through the moving, or copying and embedding, of VDE container objects. Application of content control information normally occurs securely within one or more VDE secure sub-system PPEs 650. This process may employ a VDE template that enables a user, through easy to use GUI user interface tools, to specify VDE content control information for certain or all embedded content, and which may include menu driven, user selectable and/or definable options, such as picking amongst alternative control methods (e.g. between different forms of metering) which may be represented by different icons picturing (symbolizing) different control functions and apply such functions to an increment of VDE secured content, such as an embedded object listed on an object directory display.

Extracting content from a VDE content container, or editing or otherwise creating VDE content with a VDE aware application, provides content which may be placed within a new VDE content container object for embedding into said parent VDE container, or such content may be directly placed into a previously existing content container. All of these processes may



be managed by processing VDE content control information within one or more VDE installation secure sub-systems.

5 VDE content container objects may be embedded in a parent object through control information referenced by a parent object permissions record that resolves said embedded object's location and/or contents. In this case, little or no change to the embedded object's previously existing content control information may be required. VDE securely managed content which is  
10 relocated to a certain VDE content container may be relocated through the use of VDE sub-system secure processes which may, for example, continue to maintain relocated content as encrypted or otherwise protected (e.g. by secure tamper resistant barrier 502) during a relocation/embedding process.

15 Embedded content (and/or content objects) may have been contributed by different parties and may be integrated into a VDE container through a VDE content and content control information integration process securely managed through the  
20 use of one or more secure VDE subsystems. This process may, for example, involve one or more of:

(1.) securely applying instructions controlling the embedding and/or use of said submitted content, wherein said

instructions were securely put in place, at least in part, by a content provider and/or user of said VDE container. For example, said user and/or provider may interact with one or more user interfaces offering a selection of content embedding and/or control options (e.g. in the form of a VDE template). Such options may include which, and/or whether, one or more controls should be applied to one or more portions of said content and/or the entry of content control parameter data (such a time period before which said content may not be used, cost of use of content, and/or pricing discount control parameters such as software program suite sale discounting). Once required and/or optional content control information is established by a provider and/or user, it may function as content control information which may be, in part or in full, applied automatically to certain, or all, content which is embedded in a VDE content container.

(2.) secure VDE managed negotiation activities, including the use of a user interface interaction between a user at a receiving VDE installation and VDE content control information associated with the content being submitted for embedding. For example, such associated control information may propose certain content information and the content receiver may, for example, accept, select from a plurality, reject, offer alternative control information, and/or apply conditions to the use of certain

content control information (for example, accept a certain one or more controls if said content is used by a certain one or more users and/or if the volume of usage of certain content exceeds a certain level).

5

(3.) a secure, automated, VDE electronic negotiation process involving VDE content control information of the receiving VDE content container and/or VDE installation and content control information associated with the submitted content (such as control information in a permissions record of a contributed VDE object, certain component assemblies, parameter data in one or more UDEs and/or MDEs, etc.).

10

Content embedded into a VDE content container may be embedded in the form of:

15

(1.) content that is directly, securely integrated into previously existing content of a VDE content container (said container may be a parent or embedded content container) without the formation of a new container object. Content control information associated with said content after embedding must be consistent with any pre-embedding content control information controlling, at least in part, the establishment of control information required after embedding. Content control

20

information for such directly integrated, embedded content may be integrated into, and/or otherwise comprise a portion of, control information (e.g. in one or more permissions records containing content control information) for said VDE container, and/or

5

(2.) content that is integrated into said container in one or more objects which are nested within said VDE content container object. In this instance, control information for said content may be carried by either the content control information for the parent VDE content container, or it may, for example, be in part or in full carried by one or more permissions records contained within and/or specifically associated with one or more content containing nested VDE objects. Such nesting of VDE content containing objects within a parent VDE content container may employ a number of levels, that is a VDE content container nested in a VDE content container may itself contain one or more nested VDE content containers.

10

15

VDE content containers may have a nested structure comprising one or more nested containers (objects) that may themselves store further containers and/or one or more types of content, for example, text, images, audio, and/or any other type of electronic information (object content may be specified by content control information referencing, for example, byte offset

locations on storage media). Such content may be stored, communicated, and/or used in stream (such as dynamically accumulating and/or flowing) and/or static (fixed, such as predefined, complete file) form. Such content may be derived by  
5 extracting a subset of the content of one or more VDE content containers to directly produce one or more resulting VDE content containers. VDE securely managed content (e.g. through the use of a VDE aware application or operating system having extraction capability) may be identified for extraction from each  
10 of one or more locations within one or more VDE content containers and may then be securely embedded into a new or existing VDE content container through processes executing VDE controls in a secure subsystem PPE 650. Such extraction and embedding (VDE "exporting") involves securely protecting,  
15 including securely executing, the VDE exporting processes.

A VDE activity related to VDE exporting and embedding involves performing one or more transformations of VDE content from one secure form to one or more other secure forms. Such  
20 transformation(s) may be performed with or without moving transformed content to a new VDE content container (e.g. by component assemblies operating within a PPE that do not reveal, in unprotected form, the results or other output of such transforming processes without further VDE processes governing

use of at least a portion of said content). One example of such a transformation process may involve performing mathematical transformations and producing results, such as mathematical results, while retaining, none, some, or all of the content information on which said transformation was performed. Other examples of such transformations include converting a document format (such as from a WordPerfect format to a Word for Windows format, or an SGML document to a Postscript document), changing a video format (such as a QuickTime video format to a MPEG video format), performing an artificial intelligence process (such as analyzing text to produce a summary report), and other processing that derives VDE secured content from other VDE secured content.

Figure 79 shows an example of an arrangement of commercial VDE users. The users in this example create, distribute, redistribute, and use content in a variety of ways. This example shows how certain aspects of control information associated with content may evolve as control information passes through a chain of handling and control. These VDE users and controls are explained in more detail below.

Creator A in this example creates a VDE container and provides associated content control information that includes

references (amongst other things) to several examples of possible "types" of VDE control information. In order to help illustrate this example, some of the VDE control information passed to another VDE participant is grouped into three categories in the following more detailed discussion: distribution control information, redistribution control information, and usage control information. In this example, a fourth category of embedding control information can be considered an element of all three of the preceding categories. Other groupings of control information are possible (VDE does not require organizing control information in this way). The content control information associated with this example of a container created by creator A is indicated on Figure 80 as  $C_A$ . Figure 80 further shows the VDE participants who may receive enabling control information related to creator A's VDE content container. Some of the control information in this example is explained in more detail below.

---

Some of the distribution control information in this example, control information primarily associated with creation, modification, and/or use of control information by distributors, specified by creator A includes: (a) distributors will compensate creator A for each active user of the content of the container at the rate of \$10 per user per month, (b) distributors are budgeted such that they may allow no more than 100 independent users to

gain access to such content (i.e. may create no more than 100 permissions records reflecting content access rights) without replenishing this budget, and (c) no distribution rights may be passed on in enabling control information (e.g. permissions records and associated component assemblies) created for distribution to other participants.

Some of the content redistribution control information (in this example, control information produced by a distributor within the scope permitted by a more senior participant in a chain of handling and control and passed to user/providers (in this example, user/distributors) and associated with controls and/or other requirements associated with redistribution activities by such user/distributors) specified by creator A includes: (a) a requirement that control information enabling content access may be redistributed by user/distributors no more than 2 levels, and further requires that each redistribution decrease this value by one, such that a first redistributor is restricted to two levels of redistribution, and a second redistributor to whom the first redistributor delivers permissions will be restricted to one additional level of redistribution, and users receiving permissions from the second redistributor will be unable to perform further redistribution (such a restriction may be enforced, for example, by including as one aspect of a VDE



control method associated with creating new permissions a requirement to invoke one or more methods that: (i) locate the current level of redistribution stored. for example, as an integer value in a UDE associated with such one or more methods, (ii)  
5 compare the level of redistribution value to a limiting value, and (iii) if such level of redistribution value is less than the limiting value, increment such level of redistribution value by one before delivering such a UDE to a user as an aspect of content control information associated with VDE managed content. or fail the  
10 process if such value is equal to such a limiting value), and (b) no other special restrictions are placed on redistributors.

Some of the usage control information (in this example, control information that a creator requires a distributor to  
15 provide in control information passed to users and/or user/distributors) specified by creator A may include, for example: (a) no moves (a form of distribution explained elsewhere in this document) of the content are permitted. and (b) distributors will be required to preserve (at a minimum)  
20 sufficient metering information within usage permissions in order to calculate the number of users who have accessed the container in a month and to prevent further usage after a rental has expired (e.g. by using a meter method designed to report access usages to creator A through a chain of handling and

reporting, and/or the use of expiration dates and/or time-aged encryption keys within a permissions record or other required control information).

5           Some of the extracting and/or embedding control information specified by creator A in this example may include a requirement that no extracting and/or embedding of the content is or will be permitted by parties in a chain of handling and control associated with this control information, except for users  
10           who have no redistribution rights related to such VDE secured content provided by Creator A. Alternatively, or in addition, as regards different portions of said content, control information enabling certain extraction and/or embedding may be provided along with the redistribution rights described in this example for  
15           use by user/distributors (who may include user content aggregators, that is they may provide content created by, and/or received from, different sources so as to create their own content products).

20           Distributor A in this example has selected a basic approach that distributor A prefers when offering enabling content control information to users and/or user/distributors that favors rental of content access rights over other approaches. In this example, some of the control information provided by

creators will permit distributor A to fulfill this favored approach directly, and other control structures may disallow this favored approach (unless, for example, distributor A completes a successful VDE negotiation allowing such an approach and supporting appropriate control information). Many of the control structures received by distributor A, in this example, are derived from (and reflect the results of) a VDE negotiation process in which distributor A indicates a preference for distribution control information that authorizes the creation of usage control information reflecting rental based usage rights. Such distribution control information may allow distributor A to introduce and/or modify control structures provided by creators in such a way as to create control information for distribution to users and/or user/distributors that, in effect, "rent" access rights. Furthermore, distributor A in this example services requests from user/distributors for redistribution rights, and therefore also favors distribution control information negotiated (or otherwise agreed to) with creators that permits distributor A to include such rights as an aspect of control information produced by distributor A.

In this example, distributor A and creator A may use VDE to negotiate (for example, VDE negotiate) for a distribution relationship. Since in this example creator A has produced a

VDE content container and associated control information that indicates creator A's desire to receive compensation based on rental of usage rights, and such control information further indicates that creator A has placed acceptable restrictions in redistribution control information that distributor A may use to service requests from user/distributors, distributor A may accept creator A's distribution control information without any negotiated changes.

After receiving enabling distribution control information from creator A, distributor A may manipulate an application program to specify some or all of the particulars of usage control information for users and/or user/distributors enabled by distributor A (as allowed, or not prevented, by senior control information). Distributor A may, for example, determine that a price of \$15 per month per user would meet distributor A's business objectives with respect to payments from users for creator A's container. Distributor A must specify usage control information that fulfill the requirements of the distribution control information given to distributor A by creator A. For example, distributor A may include any required expiration dates and/or time-aged encryption keys in the specification of control information in accordance with creator A's requirements. If distributor A failed to include such information (or to meet

other requirements) in their specification of control information, the control method(s) referenced in creator A's permissions record and securely invoked within a PPE 650 to actually create this control information would, in this example, fail to execute in the desired way (e.g. based on checks of proposed values in certain fields, a requirement that certain methods be included in permissions, etc.) until acceptable information were included in distributor A's control information specification.

10 In this example, user A may have established an account with distributor A such that user A may receive VDE managed content usage control information from distributor A. User A may receive content usage control information from distributor A to access and use creator A's content. Since the usage control information has passed through (and been added to, and/or modified by) a chain of handling including distributor A, the usage control information requested from distributor A to make use of creator A's content will, in this example, reflect a composite of control information from creator A and distributor A. For example, creator A may have established a meter method that will generate an audit record if a user accesses creator A's VDE controlled content container if the user has not previously accessed the container within the same calendar month (e.g. by storing the date of the user's last access in a UDE associated

with an open container event referenced in a method core of such a meter method and comparing such a date upon subsequent access to determine if such access has occurred within the same calendar month). Distributor A may make use of such a meter method in a control method (e.g. also created and/or provided by creator A, or created and/or provided by distributor A) associated with opening creator A's container that invokes one or more billing and/or budget methods created, modified, referenced in one or more permissions records and/or parameterized by distributor A to reflect a charge for monthly usage as described above. If distributor A has specified usage and/or redistribution control information within the boundaries permitted by creator A's senior control information, a new set of control information (shown as  $D_A(C_A)$  in Figure 80) may be associated with creator A's VDE content container when control information associated with that container by distributor A are delivered to users and/or user/distributors (user A, user B, and user distributor A in this example).

In this example, user A may receive control information related to creator A's VDE content container from distributor A. This control information may represent an extended agreement between user A and distributor A (e.g. regarding fees associated with use of content, limited redistribution rights, etc.) and

distributor A and creator A (e.g. regarding the character, extent, handling, reporting, and/or other aspects of the use and/or creation of VDE controlled content usage information and/or content control information received, for example, by distributor A from creator A, or vice versa, or in other VDE content usage information handling). Such an extended agreement is enforced by processes operating within a secure subsystem of each participant's VDE installation. The portion of such an extended agreement representing control information of creator A as modified by distributor A in this example is represented by  $D_A(C_A)$ , including, for example, (a) control structures (e.g. one or more component assemblies, one or more permissions records, etc.), (b) the recording of usage information generated in the course of using creator A's content in conformance with requirements stated in such control information, (c) making payments (including automatic electronic credit and/or currency payments "executed" in response to such usage) as a consequence of such usage (wherein such consequences may also include electronically, securely and automatically receiving a bill delivered through use of VDE, wherein such a bill is derived from said usage), (d) other actions by user A and/or a VDE secure subsystem at user A's VDE installation that are a consequence of such usage and/or such control information.

In addition to control information  $D_A(C_A)$ , user A may enforce her own control information on her usage of creator A's VDE content container (within the limits of senior content control information). This control information may include, for example, (a) transaction, session, time based, and/or other thresholds placed on usage such that if such thresholds (e.g. quantity limits, for example, self imposed limits on the amount of expenditure per activity parameter) are exceeded user A must give explicit approval before continuing, (b) privacy requirements of user A with respect to the recording and/or transmission of certain usage related details relating to user A's usage of creator A's content, (c) backup requirements that user A places on herself in order to help ensure a preservation of value remaining in creator A's content container and/or local store of electronic credit and/or currency that might otherwise be lost due to system failure or other causes. The right to perform in some or all of these examples of user A's control information, in some examples, may be negotiated with distributor A. Other such user specified control information may be enforced independent of any control information received from any content provider and may be set in relationship to a user's, or more generally, a VDE installation's, control information for one or more classes, or for all classes, of content and/or electronic appliance usage. The entire set of VDE control information that may be in place



during user A's usage of creator A's content container is referred to on Figure 80 as  $U_A(D_A(C_A))$ . This set may represent the control information originated by creator A, as modified by distributor A, as further modified by user A, all in accordance with control information from value chain parties providing more senior control information, and therefore constitutes, for this example, a "complete" VDE extended agreement between user A, distributor A, and creator A regarding creator A's VDE content container. User B may, for example, also receive such control information  $D_A(C_A)$  from distributor A, and add her own control information in authorized ways to form the set  $U_B(D_A(C_A))$ .

User/distributor A may also receive VDE control information from distributor A related to creator A's VDE content container. User/distributor A may, for example, both use creator A's content as a user and act as a redistributor of control information. In this example, control information  $D_A(C_A)$  both enables and limits these two activities. To the extent permitted by  $D_A(C_A)$ , user/distributor A may create their own control information based on  $D_A(C_A)$  --  $UD_A(D_A(C_A))$  -- that controls both user/distributor A's usage (in a manner similar to that described above in connection with user A and user B), and control information redistributed by user/distributor A (in a manner similar to that described above in connection with distributor A).

For example, if user/distributor A redistributes  $UD_A(D_A(C_A))$  to user/distributor B, user/distributor B may be required to report certain usage information to user/distributor A that was not required by either creator A or distributor A. Alternatively or in addition, user/distributor B may, for example, agree to pay user/distributor A a fee to use creator A's content based on the number of minutes user/distributor B uses creator A's content (rather than the monthly fee charged to user/distributor A by distributor A for user/distributor B's usage).

In this example, user/distributor A may distribute control information  $UD_A(D_A(C_A))$  to user/distributor B that permits user/distributor B to further redistribute control information associated with creator A's content. User/distributor B may make a new set of control information  $UD_B(UD_A(D_A(C_A)))$ . If the control information  $UD_A(D_A(C_A))$  permits user/distributor B to redistribute, the restrictions on redistribution from creator A in this example will prohibit the set  $UD_B(UD_A(D_A(C_A)))$  from including further redistribution rights (e.g. providing redistribution rights to user B) because the chain of handling from distributor A to user/distributor A (distribution) and the continuation of that chain from user/distributor A to user/distributor B (first level of redistribution) and the further continuation of that chain to another user represents two levels

of redistribution, and, therefore, a set  $UD_B(UD_A(D_A(C_A)))$  may not, in this example, include further redistribution rights.

As indicated in Figure 79, user B may employ content from both user/distributor B and distributor A (amongst others). In this example, as illustrated in Figure 80, user B may receive control information associated with creator A's content from distributor A and/or user/distributor B. In either case, user B may be able to establish their own control information on  $D_A(C_A)$  and/or  $UD_B(UD_A(D_A(C_A)))$ , respectively (if allowed by such control information. The resulting set(s) of control information,  $U_B(D_A(C_A))$  and/or  $U_B(UD_B(UD_A(D_A(C_A))))$  respectively, may represent different control scenarios, each of which may have benefits for user B. As described in connection with an earlier example, user B may have received control information from user/distributor B along a chain of handling including user/distributor A that bases fees on the number of minutes that user B makes use of creator A's content (and requiring user/distributor A to pay fees of \$15 per month per user to distributor A regardless of the amount of usage by user B in a calendar month). This may be more favorable under some circumstances than the fees required by a direct use of control information provided by distributor A, but may also have the disadvantage of an exhausted chain of redistribution and, for

example, further usage information reporting requirements included in  $UD_B(UD_A(D_A(C_A)))$ . If the two sets of control information  $D_A(C_A)$  and  $UD_B(UD_A(D_A(C_A)))$  permit (e.g. do not require exclusivity enforced, for example, by using a registration interval in an object registry used by a secure subsystem of user B's VDE installation to prevent deregistration and reregistration of different sets of control information related to a certain container (or registration of plural copies of the same content having different control information and/or being supplied by different content providers) within a particular interval of time as an aspect of an extended agreement for a chain of handling and control reflected in  $D_A(C_A)$  and/or  $UD_B(UD_A(D_A(C_A)))$ ), user B may have both sets of control information registered and may make use of the set that they find preferable under a given usage scenario.

In this example, creator B creates a VDE content container and associates a set of VDE control information with such container indicated in Figure 81 as  $C_B$ . Figure 81 further shows the VDE participants who may receive enabling control information related to creator B's VDE content container. In this example, control information may indicate that distributors of creator B's content: (a) must pay creator B \$0.50 per kilobyte of information decrypted by users and/or user/distributors

authorized by such a distributor, (b) may allow users and/or  
 user/distributors to embed their content container in another  
 container while maintaining a requirement that creator B  
 receive \$0.50 per kilobyte of content decrypted, (c) have no  
 5 restrictions on the number of enabling control information sets  
 that may be generated for users and/or user/distributors, (d)  
 must report information concerning the number of such  
 distributed control information sets at certain time intervals (e.g.  
 at least once per month), (e) may create control information that  
 10 allows users and/or user/distributors to perform up to three  
 moves of their control information, (f) may allow redistribution of  
 control information by user/distributors up to three levels of  
 redistribution, (g) may allow up to one move per user receiving  
 redistributed control information from a user/distributor.

15

In this example, distributor A may request control  
 information from creator B that enables distributor A to  
 distribute control information to users and/or user/distributors  
 that is associated with the VDE container described above in  
 connection with creator B. As stated earlier, distributor A has  
 20 established a business model that favors "rental" of access rights  
 to users and user/distributors receiving such rights from  
 distributor A. Creator B's distribution control information in  
 this example does not force a model including "rental" of rights.

but rather bases payment amounts on the quantity of content  
 decrypted by a user or user/distributor. In this example,  
 distributor A may use VDE to negotiate with creator B to include  
 a different usage information recording model allowed by creator  
 5 B. This model may be based on including one or more meter  
 methods in control structures associated with creator B's  
 container that will record the number of bytes decrypted by end  
 users, but not charge users a fee based on such decryptions;  
 rather distributor A proposes, and creator B's control information  
 10 agrees to allow, a "rental" model to charge users, and determines  
 the amount of payments to creator B based on information  
 recorded by the bytes decrypted meter methods and/or collections  
 of payment from users.

15 Creator B may, for example, (a) accept such a new control  
 model with distributor A acting as the auditor (e.g. trusting a  
 control method associated with processing audit information  
 received by distributor A from users of creator B's content using  
 a VDE secure subsystem at distributor A's site, and further to  
 20 securely calculate amounts owed by distributor A to creator B  
 and, for example, making payments to creator B using a  
 mutually acceptable budget method managing payments to  
 creator B from credit and/or currency held by distributor A), (b)  
 accept such a new control model based on distributor A's

acceptance of a third party to perform all audit functions associated with this content, (c) may accept such a model if information associated with the one or more meter methods that record the number of bytes decrypted by users is securely packaged by distributor B's VDE secure subsystem and is securely, employing VDE communications techniques, sent to creator B in addition to distributor A. and/or (d) other mutually acceptable conditions. Control information produced by distributor A based on modifications performed by distributor A as permitted by  $C_B$  are referred to in this example as  $D_A(C_B)$ .

User A may receive a set of control information  $D_A(C_B)$  from distributor A. As indicated above in connection with content received from creator A via a chain of handling including distributor A, user A may apply their own control information to the control information  $D_A(C_B)$ , to the extent permitted by  $D_A(C_B)$ , to produce a set of control information  $U_A(D_A(C_B))$ . The set of control information  $D_A(C_B)$  may include one or more meter methods that record the number of bytes of content from creator B's container decrypted by user A (in order to allow correct calculation of amounts owed by distributor A to creator B for user A's usage of creator B's content in accordance with the control information of  $C_B$  that requires payment of \$0.50 per kilobyte of decrypted information), and a further meter method

associated with recording usage such that distributor A may gather sufficient information to securely generate billings associated with user A's usage of creator B's content and based on a "rental" model (e.g. distributor A may, for example, have  
5 included a meter method that records each calendar month that user A makes use of creator B's content, and relates to further control information that charges user A \$10 per month for each such month during which user A makes use of such content.)

10 User/distributor A may receive control information  $C_B$  directly from creator B. In this case, creator B may use VDE to negotiate with user/distributor A and deliver a set of control information  $C_B$  that may be the same or differ from that described above in connection with the distribution relationship  
15 established between creator B and distributor A. For example, user/distributor A may receive control information  $C_B$  that includes a requirement that user/distributor A pay creator B for content decrypted by user/distributor A (and any participant receiving distributed and/or redistributed control information  
20 from user/distributor A) at the rate of \$0.50 per kilobyte. As indicated above, user/distributor A also may receive control information associated with creator B's VDE content container from distributor A. In this example, user/distributor A may have a choice between paying a "rental" fee through a chain of



handling passing through distributor A, and a fee based on the quantity of decryption through a chain of handling direct to creator B. In this case, user/distributor A may have the ability to choose to use either or both of  $C_B$  and  $D_A(C_B)$ . As indicated  
 5 earlier in connection with a chain of handling including creator A and distributor A, user/distributor A may apply her own control information to the extent permitted by  $C_B$  and/or  $D_A(C_B)$  to form the sets of control information  $UD_A(C_B)$  and  $UD_A(D_A(C_B))$ , respectively.

10

As illustrated in Figure 81, in this example, user B may receive control information associated with creator B's VDE content container from six different sources:  $C_B$  directly from creator B,  $D_A(C_B)$  from distributor A,  $UD_B(UD_A(D_A(C_B)))$  and/or  
 15  $UD_B(UD_A(C_B))$  from user/distributor B,  $D_C(C_B)$  from distributor C, and/or  $D_B(D_C(C_B))$  from distributor B. This represents six chains of handling through which user B may enter into extended agreements with other participants in this example. Two of these chains pass through user/distributor B. Based on a  
 20 VDE negotiation between user/distributor B and user B, an extended agreement may be reached (if permitted by control information governing both parties) that reflects the conditions under which user B may use one or both sets of control information. In this example, two chains of handling and control

may "converge" at user/distributor B, and then pass to user B (and if control information permits, later diverge once again based on distribution and/or redistribution by user B).

5           In this example, creator C produces one or more sets of control information  $C_C$  associated with a VDE content container created by creator C, as shown in Figure 82. Figure 82 further shows the VDE participants who may receive enabling control information related to creator C's VDE content container. The  
10       content in such a container is, in this example, organized into a set of text articles. In this example control information may include one or more component assemblies that describe the articles within such a container (e.g. one or more event methods referencing map tables and/or algorithms that describe the  
15       extent of each article).  $C_C$  may further include, for example: (a) a requirement that distributors ensure that creator C receive \$1 per article accessed by users and/or user/distributors, which payment allows a user to access such an article for a period of no more than six months (e.g. using a map-type meter method that  
20       is aged once per month, time aged decryption keys, expiration dates associated with relevant permissions records, etc.), (b) control information that allows articles from creator C's container to be extracted and embedded into another container for a one time charge per extract/embed of \$10, (c) prohibits

extracted/embedded articles from being reextracted, (d) permits distributors to create enabling control information for up to 1000 users or user/distributors per month, (e) requires that information regarding the number of users and user/distributors enabled by a distributor be reported to creator C at least once per week, (f) permits distributors to enable users or user/distributors to perform up to one move of enabling control information, and (g) permits up to 2 levels of redistribution by user/distributors.

10           In this example, distributor B may establish a distribution relationship with creator C. Distributor B in this example may have established a business model that favors the distribution of control information to users and user/distributors that bases payments to distributor B based on the number of accesses performed by such VDE participants. In this example, distributor B may create a modified set  $D_B(C_C)$  of enabling control information for distribution to users and/or user/distributors. This set  $D_B(C_C)$  may, for example, be based on a negotiation using VDE to establish a fee of \$0.10 per access per user for users and/or user/distributors who receive control information from distributor B. For example, if one or more map-type meter methods have been included in  $C_C$  to ensure that adequate information may be gathered from users and/or user/distributors to ensure correct payments to creator C by

distributor B based on  $C_C$ , such methods may be preserved in the set  $D_B(C_C)$ , and one or more further meter methods (and any other necessary control structures such as billing and/or budget methods) may be included to record each access such that the set  $D_B(C_C)$  will also ensure that distributor B will receive payments based on each access.

The client administrator in this example may receive a set of content control information  $D_B(C_C)$  that differs, for example, from control information received by user B from distributor B. For example, the client administrator may use VDE to negotiate with distributor B to establish a set of control information for content from all creators for whom distributor B may provide enabling content control information to the client administrator. For example, the client administrator may receive a set of control information  $D_B(C_C)$  that reflects the results of a VDE negotiation between the client administrator and distributor B. The client administrator may include a set of modifications to  $D_B(C_C)$  and form a new set  $CA(D_B(C_C))$  that includes control information that may only be available to users and user/distributors within the same organization as the client administrator (e.g. coworkers, employees, consultants, etc.) In order to enforce such an arrangement,  $CA(D_B(C_C))$  may, for example, include control structures that examine name services information associated

with a user or user/distributor during registration, establish a new budget method administered by the client administrator and required for use of the content, etc.

5           A distributor may provide redistribution rights to a client administrator which allows said administrator to redistribute rights to create permissions records for certain content (redistribute rights to use said content) only within the administrator's organization and to no other parties. Similarly, 10 such administrator may extend such a "limited" right to redistribute to department and/or other administrator within his organization such that they may redistribute such rights to use content based on one or more restricted lists of individuals and/or classes and/or other groupings of organization personnel 15 as defined by said administrator. This VDE capability to limit redistribution to certain one or more parties and/or classes and/or other groupings of VDE users and/or installations can be applied to content by any VDE content provider, so long as such a control is allowed by senior control information.

20

User D in this example may receive control information from either the client administrator and/or user/distributor C. User/distributor C may, for example, distribute control information  $UD_C(CA(D_B(C_C)))$  to user D that includes a

departmental budget method managed by user/distributor C to  
allow user/distributor C to maintain an additional level of control  
over the actions of user D. In this case,  $UD_C(CA(D_B(C_C)))$  may  
include multiple levels of organizational controls (e.g. controls  
5 originating with the client administrator and further controls  
originating with user/distributor C) in addition to controls  
resulting from a commercial distribution channel. In addition or  
alternatively, the client administrator may refuse to distribute  
certain classes of control information to user D even if the client  
10 administrator has adequate control information (e.g. control  
information distributed to user/distributor C that allows  
redistribution to users such as user D) to help ensure that  
control information flows through the client administrator's  
organization in accordance with policies, procedures, and/or  
15 other administrative processes.

In this example, user E may receive control information  
from the client administrator and/or distributor B. For example,  
user E may have an account with distributor B even though  
20 some control information may be received from the client  
administrator. In this case, user E may be permitted to request  
and receive control information from distributor B without  
restriction, or the client administrator may have, as a matter of  
organizational policy, control information in place associated

with user E's electronic appliance that limits the scope of user E's interaction with distributor B. In the latter case, the client administrator may, for example, have limited user E to registering control information with the secure subsystem of user E's electronic appliance that is not available from the client administrator, is from one or more certain classes of distributors and/or creators, and/or has a cost for usage, such as a certain price point (e.g. \$50 per hour of usage). Alternatively or in addition, the client administrator may, for example, limit user E to receiving control information from distributor B in which user E receives a more favorable price (or other control information criteria) than the price (or other criteria) available in control information from the client administrator.

15 In this example, creator D may create a VDE content container that is designed primarily for integration with other content (e.g. through use of a VDE extracting/embedding process), for example, content provided by creator B and creator C. Figure 83 shows the VDE participants who may receive enabling control information related a VDE content container produced by creator D. Control information associated with creator D's content ( $C_D$  in Figure 83) may include, for example:

20 (a) a requirement that distributors make payment of either \$1.50 per open per user, or \$25 per user for an unlimited number of

opens, (b) a discount of 20% for any user that has previously paid for an unlimited number of opens for certain other content created by creator D (e.g. implemented by including one or more billing methods that analyze a secure database of a user's VDE installation to determine if any of such certain other containers are registered, and further determines the character of rights held by a user purchasing rights to this container), (c) a requirement that distributors report the number of users and user/distributors enabled by control information produced in accordance with C<sub>D</sub> after such number exceeds 1000, (d) a requirement that distributors limit the number of moves by users and/or user/distributors to no more than one, (e) a requirement that distributors limit user/distributors to no more than four levels of redistribution, and (f) that distributors may create enabling control information that permits other distributors to create control information as distributors, but may not pass this capability to such enabled distributors, and further requires that audit information associated with use of control information by such enabled distributors shall pass directly to creator D without processing by such enabling distributor and that creator D shall pay such an enabling distributor 10% of any payments received by creator D from such an enabled distributor.



In this example, distributor C may receive VDE content containers from creator B, creator C, and creator D, and associated sets of control information  $C_B$ ,  $C_C$ , and  $C_D$ .

5 Distributor C may use the embedding control information and other control information to produce a new container with two or more VDE objects received from creator B, creator C, and creator D. In addition or alternatively, distributor C may create enabling control information for distribution to users and/or user/distributors (or in the case of  $C_D$ , for distributors) for such  
10 received containers individually. For example, distributor C may create a container including content portions (e.g. embedded containers) from creator B, creator C, and creator D in which each such portion has control information related to its access and use that records, and allows an auditor to gather, sufficient  
15 information for each such creator to securely and reliably receive payments from distributor C based on usage activities related to users and/or user/distributors enabled by distributor C.

Furthermore, distributor C may negotiate using VDE with some or all of such creators to enable a model in which distributor C  
20 provides overall control information for the entire container based on a "uniform" fee (e.g. calculated per month, per access, from a combined model, etc.) charged to users and/or user/distributors, while preserving the models of each such creator with respect to payments due to them by distributor C

based on  $C_B$ ,  $C_C$ , and/or  $C_D$ , and, for example, resulting from each of their differing models for the collection of content usage information and any related (e.g. advertising) information.

5           In this example, distributor B may receive a VDE content container and associated content control information  $C_E$  from creator E as shown in Figure 83. If  $C_E$  permits, distributor B may extract a portion of the content in such a container. Distributor B may then, for example, embed this portion in a  
10           container received from distributor C that contains an aggregation of VDE objects created by creator B, creator C, and creator D. Depending on the particular restrictions and/or permissions in the sets of control information received from each creator and distributor C, distributor B may, for example, be able  
15           to embed such an extracted portion into the container received from distributor C as an independent VDE object, or directly into content of "in place" objects from creator B, creator C, and/or creator D. Alternatively, or in addition, distributor B may, if permitted by  $C_E$ , choose to distribute such an extracted portion  
20           of content as an independent VDE object.

          User B may, in this example, receive a VDE content container from distributor C that is comprised of VDE objects created by creator B, creator C, and creator D. In addition, user

B may receive a VDE content container from distributor B that contains the same content created by creator B, creator C, and creator D in addition to one or more extracted/embedded portions of content created by creator E. User B may base decisions  
5 concerning which of such containers they choose to use (including which embedded containers she may wish to use), and under which circumstances, based on, for example, the character of such extracted/embedded portions (e.g. multimedia presentations illustrating potential areas of interest in the  
10 remainder of the content, commentary explaining and/or expositing other elements of content, related works, improved application software delivered as an element of content, etc.); the quality, utility, and/or price (or other attributes of control information) of such portions; and other considerations which  
15 distinguish the containers and/or content control information received, in this example, from distributor B and distributor C.

User B may receive content control information from distributor B for such a VDE content container that permits user  
20 B to add and/or modify content contained therein. User B may, for example, desire an ability to annotate content in such a container using a VDE aware word processor or other application(s). If permitted by senior control information, some or all of the content may be available to user B for modification

and/or additions. In this case, user B is acting as a VDE creator for added and/or modified content. User B may, for example, provide new control information for such content, or may be required (or desire to) make use of existing control information (or control information included by senior members of a chain of handling for this purpose) to manage such content (based on control information related to such a container and/or contained objects).

10           In this example, VDE 100 has been used to enable an environment including, for example, content distribution, redistribution, aggregation (extracting and/or embedding), reaggregation, modification, and usage. The environment in this example allows competitive models in which both control  
15           information and content may be negotiated for and have different particulars based on the chain of handling through which control information and/or content has been passed. Furthermore, the environment in this example permits content to be added to, and/or modified by, VDE participants receiving  
20           control information that enables such activities.

### **Example -- Content Distribution Through a Content VDE Chain of Handling**

Figure 84 reflects certain aspects of a relatively simple model 3400 of VDE content distribution involving several categories of VDE participants. In this instance, and for simplicity of reference purposes, various portions of content are represented as discrete items in the form of VDE content container objects. One or more of such content portions may also be integrated together in a single object and may (as may the contents of any VDE content container object if allowed by content control information) be extracted in whole or part by a user. In this example, publishers of historical/educational multimedia content have created VDE content containers through the use of content objects available from three content resources:

- a Video Library 3402 product available to Publishers on optical discs and containing video clip VDE objects representing various historical situations,
- an Internet Repository 3404 which stores history information text and picture resources in VDE objects which are available for downloading to Publishers and other users, and

- an Audio Library 3406, also available on optical discs, and containing various pieces of musical performances and vocal performances (for example, historical narrations) which can be used alone or to accompany other educational historical materials.

The information provided in library 3402, repository 3404, and library 3406 may be provided to different publishers 3408(a), 3408(b), ..., 3408(n). Publishers 3408 may, in turn, provide some or all of the information they obtain to end users 3410.

In this example, the Video Library 3402 control information allows publishers to extract objects from the Video Library product container and content control information enabling use of each extracted object during a calendar year if the object has a license cost of \$50 or less, and is shorter than 45 minutes in duration, and 20,000 copies of each of any other extracted objects, and further requires all video objects to be VDE fingerprinted upon decryption. The Audio Library 3404 has established similar controls that match its business model. The Internet Repository 3406 VDE containerizes, including encrypts, selected object content as it streams out of the Repository in response to an online, user request to download an object. The Repository 3406 may fingerprint the identification of the

receiving VDE installation into its content prior to encryption and communication to a publisher, and may further require user identification fingerprinting of their content when decrypted by said Publisher or other content user.

5

The Publishers 3408 in this example have selected, under terms and conditions VDE negotiated (or otherwise agreed to) with the providing resources, various content pieces which they combine together to form their VDE object container products for their teacher customers. Publisher 3408(A) has combined video objects extracted from the Video Library 3402 (as indicated by circles), text and image objects extracted from the Internet Repository 3404 (indicated by diamonds), and one musical piece and one historical narration extracted from the Audio Library 3406 (as indicated by rectangles). Publisher 3408(B) has extracted a similar array of objects to be combined into his product, and has further added graphical elements (indicated by a hexagon) created by Publisher 3408(B) to enhance the product. Publisher 3408(C) has also created a product by combining objects from the Internet Repository 3404 and the Audio Library 3406. In this example, all publisher products are delivered, on their respective optical discs, in the form of VDE content container objects with embedded objects, to a modern high school for installation on the high school's computer network.

In this particular example, End-Users 3410 are teachers who use their VDE node's secure subsystems to access the VDE installation on their high school server that supports the publishers' products (in an alternative example, the high school may maintain only a server based VDE installation). These teachers license the VDE products from one or more of the publishers and extract desired objects from the VDE product content containers and either download the extracted VDE content in the form of VDE content containers for storage on their classroom computers and/or as appropriate and/or efficient. The teachers may store extracted content in the form of VDE content containers on server mass storage (and/or if desired and available to an end-user, and further according to acceptable pricing and/or other terms and conditions and/or senior content control information, they may store extracted information in "clear" unencrypted form on their nodes' and/or server storage means). This allows the teachers to play, and/or otherwise use, the selected portions of said publishers' products, and as shown in two instances in this example, add further teacher and/or student created content to said objects. End-user 3410(2), for example, has selected a video piece 1 received from Publisher A, who received said object from the Video Library. End-user 3410(3) has also received a video piece 3 from the same Publisher 3408(A) wherein said piece was also available to her from



Publisher 3408(B), but perhaps under not as favorable terms and conditions (such as a support consultation telephone line). In addition, end-user 3410(3) has received an audio historical narration from Publisher 3408(B) which corresponds to the content of historical reference piece 7. End-user 3410(3) has also received a corresponding historical reference piece 7 (a book) from publisher 3408(2) who received said book from the Internet Repository 3404. In this instance, perhaps publisher 3408(2) charged less for said book because end-user 3410(3) has also licensed historical reference piece 7 from him, rather than publisher 3408(1), who also carried the same book. End-user 3410(3), as a teacher, has selected the items she considers most appropriate for her classes and, through use of VDE, has been able to flexibly extract such items from resources available to her (in this instance, extracting objects from various optical products provided by publishers and available on the local high school network server).

**Example -- Distribution of Content Control Information Within an Organization**

Figure 85 shows two VDE content containers, Container 300(A) and Container 300(B), that have been distributed to a VDE Client Administrator 3450 in a large organization. As shown in the figure, Container 300(A) and Container 300(B), as

they arrive at the corporation, carry certain control information specifying available usage rights for the organization. As can be further seen in Figure 85, the client administrator 3450 has distributed certain subsets of these rights to certain department administrators 3452 of her organization, such as Sales and Marketing Administrator 3452(1), Planning Administrator 3452(2), and Research and Development Administrator 3452(k). In each instance, the Client Administrator 3450 has decided which usage options and how much budget should be made available to each department.

Figure 85 is a simplified example and, for example, the Client Administrator 3450 could have added further VDE controls created by herself and/or modified and/or deleted in place controls (if allowed by senior content control information) and/or (if allowed by control information) she could have further divided the available monetary budget (or other budgets) among specific usage activities. In this example, departmental administrators have the same rights to determine the rights of departmental end-users as the client administrator has in regard to departments. In addition, in this example (but not shown in Figure 85) the client administrator 3450 and/or content provider(s) may also determine certain control information which must directly control (including providing rights related to) end-

user content usage and/or the consequences of said usage for all or certain classes of end-users. In the example shown in Figure 85, there are only three levels of VDE participants within the organization:

- 5           a Client Administrator 3450,  
          department administrators 3452, and  
          end-users 3454.

10           In other examples, VDE will support many levels of VDE  
          administration (including overlapping groups) within an  
          organization (e.g., division, department, project, network, group,  
          end-users, etc). In addition, administrators in a VDE model may  
          also themselves be VDE content users.

15           Within an organization, VDE installations may be at each  
          end-user 3454 node, only on servers or other multiple user  
          computers or other electronic appliances, or there may be a  
          mixed environment. Determination as to the mix of VDE server  
          and/or node usage may be based on organization and/or content  
          provider security, performance, cost overhead, or other  
20           considerations.

          In this example, communications between VDE  
          participants in Figure 85 employs VDE secure communication  
          techniques between VDE secure subsystems supporting PPEs

and other VDE secure system components at each VDE installation within the organization.

**Example -- Another Content Distribution Example**

5           Creators of VDE protected content may interact with other VDE participants in many different ways. A VDE creator 102 may, for example, distribute content and/or content control information directly to users, distribute content and/or content control information to commercial content repositories, distribute  
10           content and/or content control information to corporate content repositories, and/or distribute content and/or content control information to other VDE participants. If a creator 102 does not interact directly with all users of her content, she may transmit distribution permissions to other VDE participants that permit  
15           such participants to further distribute content and/or content control information. She may also allow further distribution of VDE content and/or content control information by, for example, not restricting redistribution of control information, or allowing a VDE participant to act as a "conduit" for one or more permissions  
20           records that can be passed along to another party, wherein said permissions record provides for including the identification of the first receiving party and/or the second receiving party.

Figure 86 shows one possible arrangement of VDE participants. In this example, creator 102 may employ one or more application software programs and one or more VDE secure subsystems to place unencrypted content into VDE protected form (i.e., into one or more VDE content containers). In addition, creator 102 may produce one or more distribution permissions 3502 and/or usage permissions 3500 as an aspect of control information associated with such VDE protected content. Such distribution and/or usage permissions 3500, 3502 may be the same (e.g., all distribution permissions may have substantively all the same characteristics), or they may differ based on the category and/or class of participant for whom they are produced, the circumstances under which they are requested and/or transmitted, changing content control models of either creator 102 or a recipient, etc.

In this example, creator 102 transmits (e.g., over a network, via broadcast, and/or through transfer of physical media) VDE protected content to user 112a, user 112b, and/or user 112c. In addition, creator 102 transmits, using VDE secure communications techniques, usage permissions to such users. User 112a, user 112b, and user 112c may use such VDE protected content within the restrictions of control information specified by usage permissions received from creator 102. In this

case, creator 102 may, for example, manage all aspects of such users activities related to VDE protected content transmitted to them by creator 102. Alternatively, creator 102 may, for example, include references to control information that must be available to users that is not provided by creator 102 (e.g., component assemblies managed by another party).

Commercial content repository 200g, in this example, may receive VDE protected (or otherwise securely delivered) content and distribution, permissions and/or other content usage control information from creator 102. Commercial content repository 200g may store content securely such that users may obtain such, when any required conditions are met, content from the repository 200g. The distribution permissions 3502 may, for example, permit commercial content repository 200g to create redistribution permissions and/or usage permissions 3500, 3502 using a VDE protected subsystem within certain restrictions described in content control information received from creator 102 (e.g., not to exceed a certain number of copies, requiring certain payments by commercial content repository 200g to creator 102, requiring recipients of such permissions to meet certain reporting requirements related to content usage information, etc.). Such content control information may be stored at the repository installation and be applied to

unencrypted content as it is transmitted from said repository in response to a user request, wherein said content is placed into a VDE container as a step in a secure process of communicating such content to a user. Redistribution permissions may, for example, permit a recipient of such permissions to create a certain number of usage permissions within certain restrictions (e.g., only to members of the same household, business other organization, etc.). Repository 200g may, for example, be required by control information received from creator 102 to gather and report content usage information from all VDE participants to whom the repository has distributed permissions.

In this example, power user 112d may receive VDE protected content and redistribution permissions from commercial content repository 200g using the desktop computer 3504. Power user 112d may, for example, then use application software in conjunction with a VDE secure subsystem of such desktop computer 3504 in order to produce usage permissions for the desktop computer 3504, laptop computer 3506 and/or settop appliance 3508 (assuming redistribution permissions received from commercial content repository 200g permit such activities). If permitted by senior control information (for example, from creator 102 as may be modified by the repository 200g), power user 112d may add her own restrictions to such usage

permissions (e.g., restricting certain members of power user 112d's household using the settop appliance to certain times of day, amounts of usage, etc. based on their user identification information). Power user 112d may then transmit such VDE  
5 protected content and usage permissions to the laptop computer 3506 and the settop appliance 3508 using VDE secure communications techniques. In this case, power user 112d has redistributed permissions from the desktop computer 3504 to the settop appliance 3508 and the laptop computer 3506, and  
10 periodically the settop appliance and the laptop computer may be required to report content usage information to the desktop computer, which in turn may aggregate, and/or otherwise process, and report user usage information to the repository 200g.

15

User 112e and/or user 112f may receive usage permissions and VDE protected content from commercial content repository 200g. These users may be able to use such content in ways authorized by such usage information. In contrast to power user  
20 112d, these users may not have requested and/or received redistribution permissions from the repository 200g. In this case, these users may still be able to transfer some or all usage rights to another electronic appliance 600, and/or they may be permitted to move some of their rights to another electronic



appliance, if such transferring and/or moving is permitted by the usage permissions received from the repository 200g. In this case, such other appliances may be able to report usage information directly to the repository 200g.

5

In this example, corporate content repository 702 within corporation 700 may receive VDE protected content and distribution permissions from creator 102. The distribution permissions received by corporate repository 702 may, for example, include restrictions that limit repository 702 to distribution activities within corporation 700.

10

The repository 702 may, for example, employ an automated system operating in conjunction with a VDE secure subsystem to receive and/or transmit VDE protected content, and/or redistribution and/or usage permissions. In this case, an automated system may, for example, rely on criteria defined by corporate policies, departmental policies, and/or user preferences to determine the character of permissions and/or content delivered to various parties (corporation groups and/or individuals) within corporation 700. Such a system may, for example, automatically produce redistribution permissions for a departmental content repository 704 in response to corporation

15

20

700 receiving distribution permissions from creator 102, and/or produce usage permissions for user 112j and/or user 112k.

5           The departmental repository 704 may automatically produce usage permissions for user 112g, user 112h, and/or user 112i. Such users may access content from the corporate content repository 702, yet receive usage permissions from departmental repository 704. In this case, user 112g, user 112h, and/or user 112i may receive usage permissions from departmental repository 704 that incorporate departmental restrictions in addition to restrictions imposed by senior control information (in this example, from creator 102, as may be modified by corporate repository 702, as may be further modified by departmental repository 704, that reflect a VDE extended agreement incorporating commercial requirements of creator 102 and corporation 700 in addition to corporate and/or departmental policies and agreements with corporate personnel of corporation 700).

20       **Example—"Virtual Silicon Container"**

As discussed above, VDE in one example provides a "virtual silicon container" ("virtual black box") in that several different instances of SPU 500 may securely communicate together to provide an overall secure hardware environment that

"virtually" exists at multiple locations and multiple electronic appliances 600. Figure 87 shows one model 3600 of a virtual silicon container. This virtual container model 3600 includes a content creator 102, a content distributor 106, one or more content redistributors 106a, one or more client administrators 700, one or more client users 3602, and one or more clearinghouses 116. Each of these various VDE participants has an electronic appliance 600 including a protected processing environment 655 that may comprise, at least in part, a silicon-based semiconductor hardware element secure processing unit 500. The various SPUs 500 each encapsulate a part of the virtual distribution environment, and thus, together form the virtual silicon container 3600.

#### 15      **Example -- Testing/Examinations**

          A scheduled SAT examination for high school seniors is prepared by the Educational Testing Service. The examination is placed in a VDE container for scheduled release on November 15, 1994 at 1:00 PM Eastern Standard time. The SAT prepares one copy of the container for each school or other location which will conduct the examination. The school or other location ("test site") will be provided with a distributed examination container securely containing the VDE identification for the "administration" electronic appliance and/or test administrator

at the test site (such as, a testing organization) and a budget enabling, for example, the creation of 200 test VDE content containers. Each container created at the test site may have a permissions record containing secure identification information for each electronic appliance 600, on the test site's network, that will be used by a test taker, as well as, for example, an identification for the student who will take the test. The student identification could, for example, be in the form of a secure PIN password which is entered by the student prior to taking the test (a test monitor or administrator might verify the student identification by entering in a PIN password). Of course, identification might take the form of automated voice recognition, handwriting recognition (signature recognition), fingerprint information, eye recognition, or similar one or more recognition forms which may be used either to confirm the identity of the test taker (and/or test monitor/administrator) and/or may be stored with the test results in a VDE container or the like or in a location pointed to by certain container information. This identification may be stored in encrypted or unencrypted form. If stored in encrypted or otherwise protected form, certain summary information, such as error correction information, may be stored with the identification information to authenticate the associated test as corresponding to the identification.

As the student takes the test using the computer terminal, the answers selected may be immediately securely stored (but may be changed by the student during the test session). Upon the completion of the test, the student's answers, along with a reference to the test, are securely stored in a VDE reporting object which is passed along to the network to the test administrator and the administration electronic appliance 600. All test objects for all students could then be placed in a VDE object 300 for communication to the Educational Testing Service, along with whatever other relevant information (which may also be secured by VDE 100), including summary information giving average and mean scores, and other information that might be desirable to summarize and/or act as an authentication of the test objects sent. For example, certain information might be sent separately from each student summary object containing information which helps validate the object as an "authentic" test object.

Applying VDE to testing scenarios would largely eliminate cheating resulting from access to tests prior to testing (normally the tests are stolen from a teacher or test administrator). At ETS, individuals who have access to tests could be limited to only a portion of the test to eliminate the risk of the theft of a "whole" test. Employing VDE would also ensure against processing

errors or other manipulation of test answers, since absolutely authentic test results can be archived for a reasonable period of time.

5                   Overall, employing VDE 100 for electronic testing will enable the benefits of electronic testing to be provided without the substantial risks associated with electronic storing, communicating, and processing of test materials and testing results. Electronic testing will provide enormous efficiency  
10 improvements, significantly lowering the cost of conducting and processing tests by eliminating printing, shipping, handling, and human processing of tests. At the same time, electronic testing will allow users to receive a copy (encrypted or unencrypted) of their test results when they leave the test sessions. This will  
15 help protect the tested individual against lost of, or improperly processed, test results. Electronic testing employing VDE 100 may also ensure that timing related variables of testing (for example precise starting, duration, and stopping times) can be reliably managed. And, of course, proper use of VDE 100 for the  
20 testing process can prevent improper access to test contents prior to testing and ensure that test taking is properly audited and authenticated, that is which person took which test, at which time, on which electronic appliance, at which location. Retesting

due to lost, stolen, improperly timed, or other variables can be avoided or eliminated.

VDE assisted testing may, of course, be employed for many  
5 different applications including secure identification of  
individuals for security/authentication purposes, for employment  
(e.g. applying for jobs) applications, and for a full range of  
evaluation testing. For example, an airline pilot, or a truck,  
train, or bus driver might take a test immediately prior to  
10 departure or during travel, with the test evaluating alertness to  
test for fatigue, drug use, etc. A certain test may have a different  
order and/or combination of test activities each time, or each  
group of times, the test is taken. The test or a master test might  
be stored in a VDE container (the order of, and which, test  
15 questions might be determined by a process executed securely  
within an PPE 650). The test responses may be encrypted as  
they occur and either locally stored for aggregated (or other test  
result) transmission or dynamically transmitted (for example, to  
a central test administration computer). If the test taker  
20 "flunks" the test, perhaps he or she is then prevented from  
operating the vehicle, either by a local PPE 650 issuing control  
instructions to that effect on some portion of the vehicle's  
electronic control system or a local PPE failing to decrypt or

otherwise provide certain key information required for vehicle operation.

### **Example -- Appliance Rental**

5           Through use of the present invention, electronic appliances can be "leased" or otherwise provided to customers who, rather than purchasing a given appliance for unlimited usage, may acquire the appliance (such as a VCR, television, microwave oven, etc.) and be charged according to one or more aspects of  
10           use. For example, the charge for a microwave might be for each time it is used to prepare an item and/or for the duration of time used. A telephone jack could be attached, either consistently or periodically, to an inexpensive modem operatively attached or within the microwave (the modem might alternatively be located  
15           at a location which services a plurality of items and/or functions -- such as burglar alarm, light and/or heat control). Alternatively, such appliances may make use of a network formed by the power cables in a building to transmit and receive signals.

20           At a periodic interval, usage information (in summary form and/or detailed) could be automatically sent to a remote information utility that collects information on appliance usage (the utility might service a certain brand, a certain type of appliance, and/or a collection of brands and/or types). The usage



information would be sent in VDE form (e.g. as a VDE object 300). The information utility might then distribute information to financial clearinghouse(s) if it did not itself perform the billing function, or the information "belonging" to each appliance

5 manufacturer and/or lessor (retailer) might be sent to them or to their agents. In this way a new industry would be enabled of leased usage of appliances where the leases might be analogous to car leasing.

10 With VDE installed, appliances could also be managed by secure identification (PIN, voice or signature recognition, etc.). This might be required each time a unit is used, or on some periodic basis. Failure to use the secure identification or use it on a timely basis could disable an appliance if a PPE 650 issued

15 one or more instructions (or failed to decrypt or otherwise provide certain information critical to appliance operation) that prevented use of a portion or all of the appliance's functions.

This feature would greatly reduce the desirability of stealing an electronic appliance. A further, allied use of VDE is the

20 "registration" of a VDE secure subsystem in a given appliance with a VDE secure subsystem at some control location in a home or business. This control location might also be responsible for VDE remote communications and/or centralized administration (including, for example, restricting your children from viewing R

rated movies either on television or videocassettes through the recognition of data indicating that a given movie, song, channel, game, etc. was R rated and allowing a parent to restrict viewing or listening). Such a control location may, for example, also  
5 gather information on consumption of water, gas, electricity, telephone usage, etc. (either through use of PPEs 650 integrated in control means for measuring and/or controlling such consumption, or through one or more signals generated by non-VDE systems and delivered to a VDE secure subsystem, for  
10 example, for processing, usage control (e.g. usage limiting), and/or billing), transmit such information to one or more utilities, pay for such consumption using VDE secured electronic currency and/or credit, etc.

15 In addition, one or more budgets for usage could be managed by VDE which would prevent improper, excessive use of a certain, leased appliance, that might, for example lead to failure of the appliance, such as making far more copies using a photocopier than specified by the duty cycle. Such improper use  
20 could result in a message, for example on a display panel or television screen, or in the form of a communication from a central clearinghouse, that the user should upgrade to a more robust model.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the  
5 contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

**WE CLAIM:**

1. A rights management appliance including:
  - a user input device,
  - 5 a user display device,
  - at least one processor, and
  - at least one element defining a protected processing environment,
  - characterized in that the protected processing environment
  - 10 stores and uses permissions, methods, keys, programs and/or
  - other information to electronically manage rights.
2. In a rights management appliance including:
  - a user input device,
  - 15 a user display device,
  - at least one processor, and
  - at least one element defining a protected processing environment,
  - a method of operating the appliance characterized by the
  - 20 step of storing and using permissions, methods, keys, programs
  - and/or other information to electronically manage rights.
3. A rights management appliance including at least one processor element at least in part defining a protected processing

environment, characterized in that the protected processing environment stores and uses permissions, methods, keys, programs and/or other information to electronically manage rights.

5

4. In a rights management appliance including at least one processor element at least in part defining a protected processing environment, a method comprising storing and using permissions, methods, keys, programs and/or other information to electronically manage rights.

10

5. An electronic appliance arrangement containing at least one secure processing unit and at least one secure database operatively connected to at least one of said secure processing unit(s), said arrangement including means to monitor usage of at least one aspect of appliance usage and control said usage based at least in part upon protected appliance usage control information.

15

6. In an electronic appliance arrangement containing at least one secure processing unit and at least one secure database operatively connected to at least one of said secure processing unit(s), a method characterized by the steps of monitoring usage of at least one aspect of appliance usage and controlling said

20

usage based at least in part upon protected appliance usage control information.

5 7. An electronic appliance arrangement containing a protected processing environment and at least one secure database operatively connected to said protected processing environment, said arrangement including means to monitor usage of at least one aspect of an amount of appliance usage and control said usage based at least in part upon protected appliance usage control information processed at least in part through use of said protected processing environment.

15 8. In an electronic appliance arrangement containing a protected processing environment and at least one secure database operatively connected to said protected processing environment, a method characterized by the steps of monitoring usage of at least one aspect of appliance usage and controlling said usage based at least in part upon protected appliance usage control information processed at least in part through use of said protected processing environment.

20 9. An electronic appliance arrangement containing one or more CPUs wherein at least one of the CPUs incorporates an integrated secure processing unit, said arrangement storing

protected appliance usage control information designed to be securely processed by said integrated secure processing unit.

5 10. In an electronic appliance arrangement containing one or more CPUs wherein at least one of the CPUs incorporates an integrated secure processing unit, a method including the step of storing and securely processing protected modular component appliance usage control information with said integrated secure processing unit.

10

11. A method of compromising a distributed electronic rights management system comprising plural nodes having protected processing environments, characterized by the following steps:

- 15 (a) exposing a certification private key,
- (b) passing at least one challenge/response protocol and/or exposing at least one external communication key based at least in part on the key exposed by the exposing step,
- (c) creating a processing environment based at least in
- 20 part on steps (a) and (b), and
- participating in distributed rights management using the processing environment created by step (c).

12. A processing environment for compromising a distributed electronic rights management system comprising plural nodes having protected processing environments, characterized by the following:

5            protocol passing means including an exposed certification private key for passing at least one challenge/response protocol, means coupled to the protocol passing means for at least one of (a) defeating an initialization challenge/response security, and/or (b) exposing external communication keys, and  
10           means coupled to the security detecting means for participating in distributed rights management.

13. A method of compromising a distributed electronic rights management system comprising plural nodes having associated  
15           protected processing environments, characterized by the steps of: compromising the permissions record of an electronic container, and using the compromised permissions record to access and/or use electronic information.

20

14. A system for compromising a distributed electronic rights management system comprising plural nodes having associated protected processing environments, characterized by means for



using a compromised permissions record of an electronic container for accessing and/or using electronic information.

- 15      15.    A method of tampering with a protected processing environment characterized by the steps of:
- discovering at least one system-wide key, and  
             using the key to obtain access to content and/or administrative information without authorization.
- 10      16.    An arrangement including means for using at least one compromised system-wide key to decrypt and compromise content and/or administrative information of a protected processing environment without authorization.
- 15      17.    A combination general and secure processing computation element comprising:
- a central processing unit;  
             at least one secure resource; and  
             a secure mode interface switch coupled between a centra
- 20      processing unit and the secure resource, the switch operable alternately in a secure mode and in a non secure mode, the switch blocking access by a central processing unit to the secure resource except when the switch is operating in the secure mode.

18. A secure printing method comprising:
- downloading a decryption program to an intelligent printer;
  - sending an encrypted print stream to the printer;
  - 5 decrypting the encrypted print stream within the printer using the decryption program; and
  - destroying the downloaded decryption program.

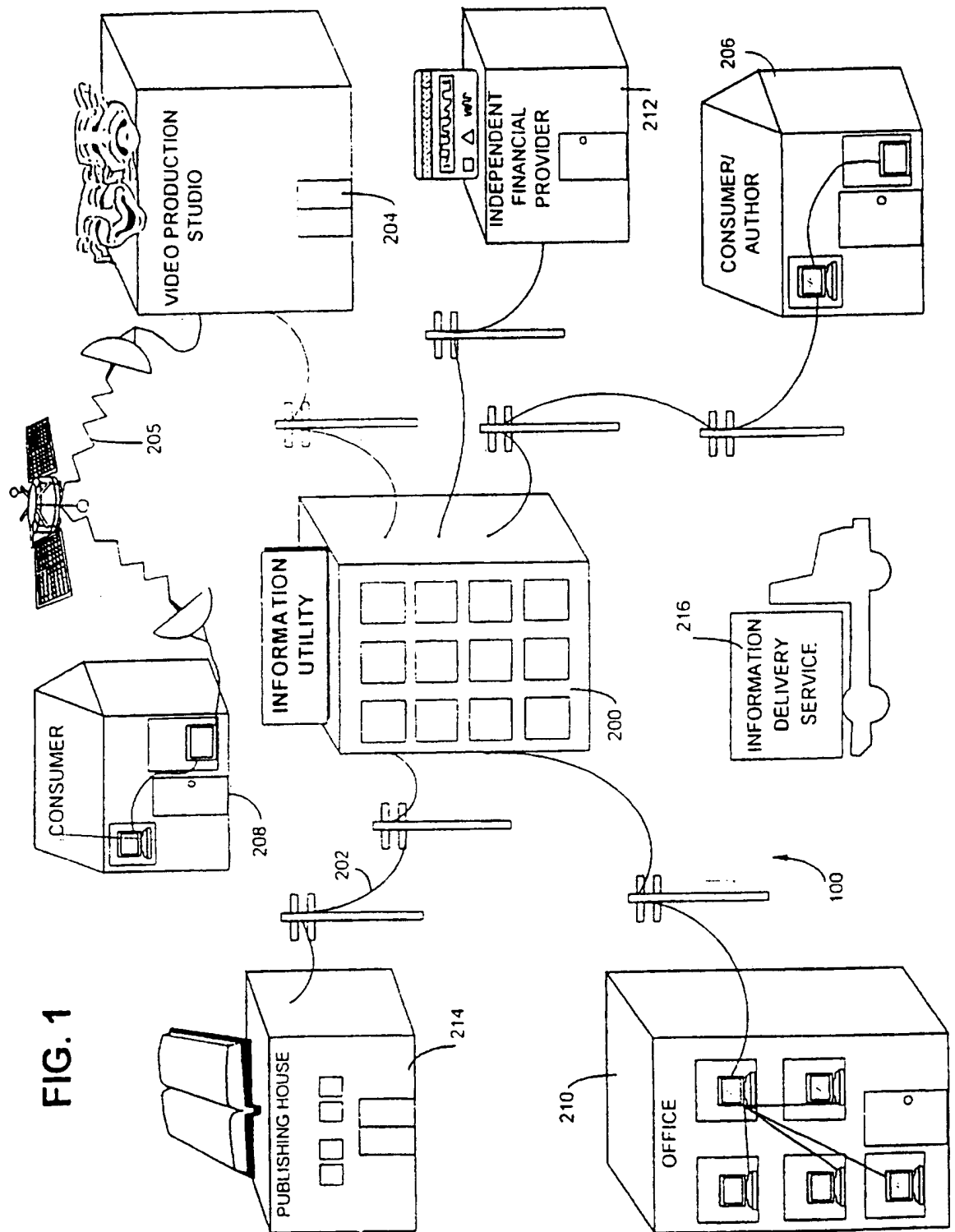


FIG. 1

2/163

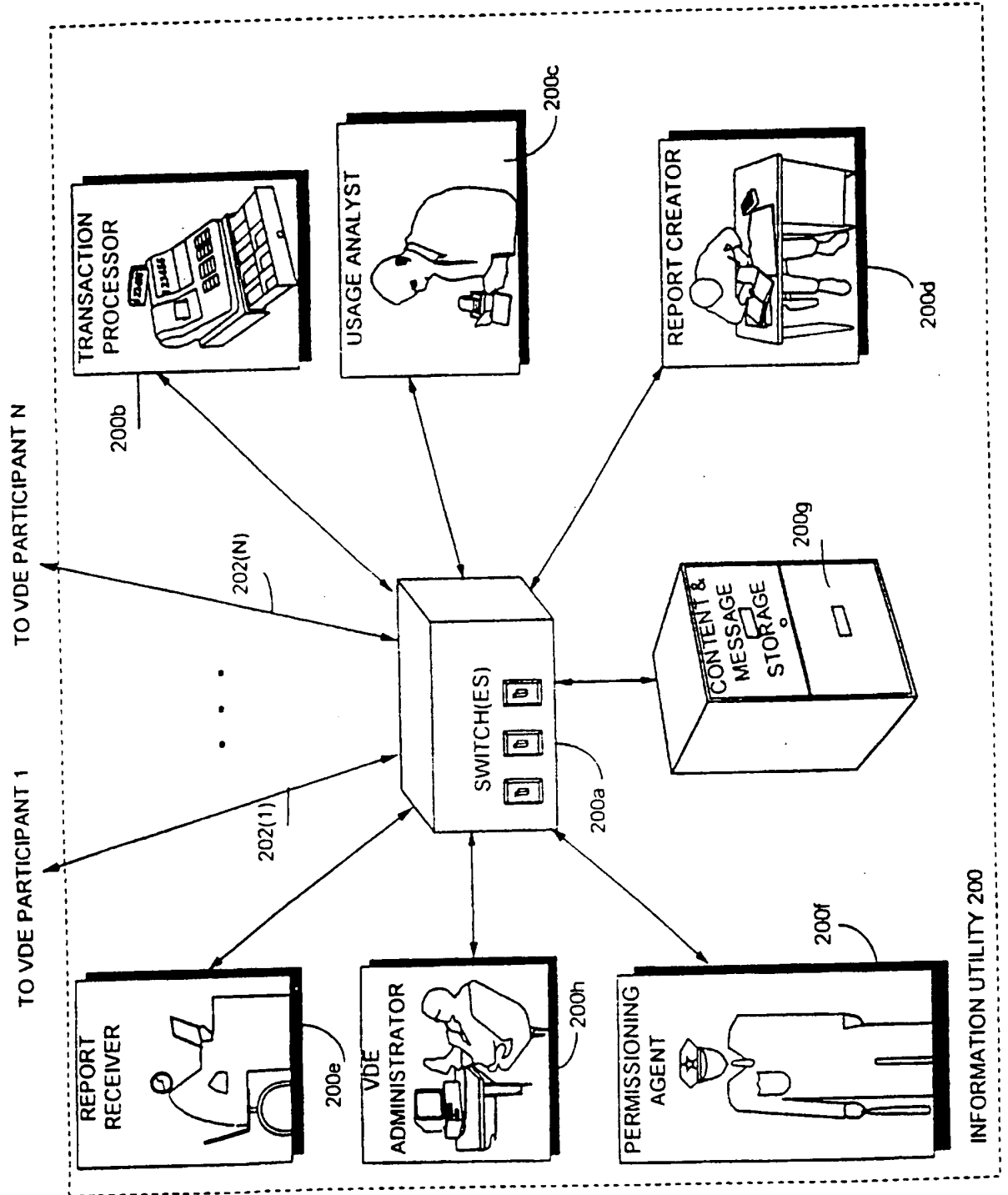
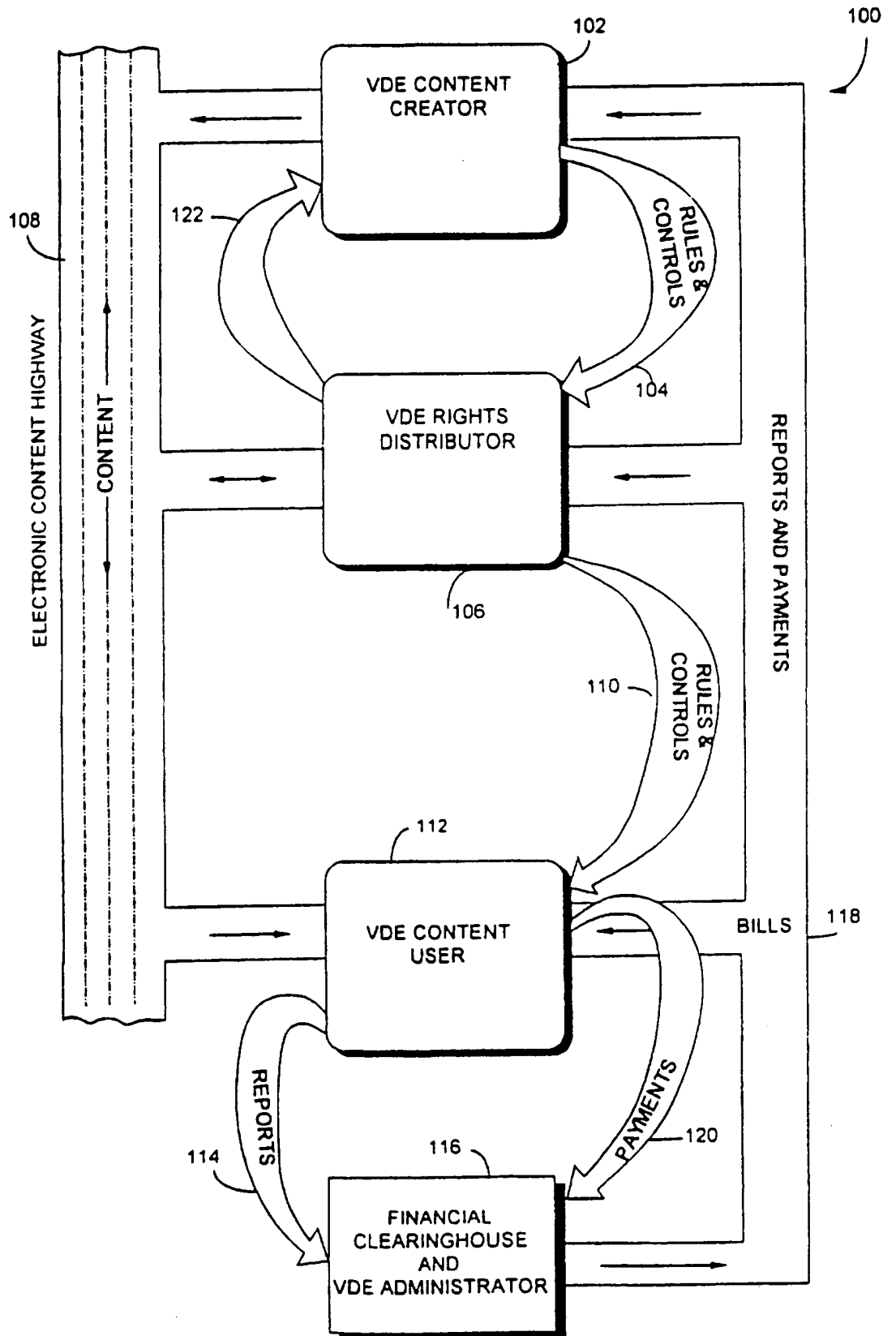


FIG. 1A

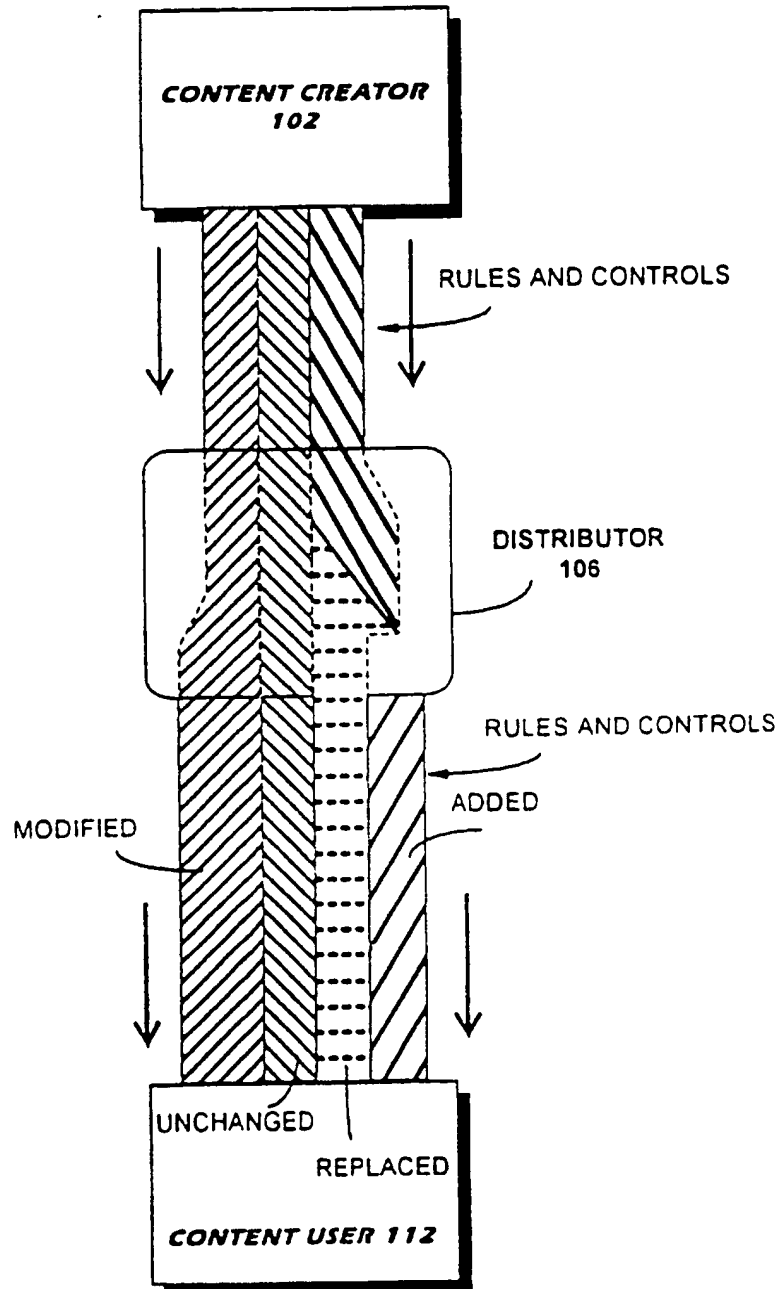
3/163

FIG. 2



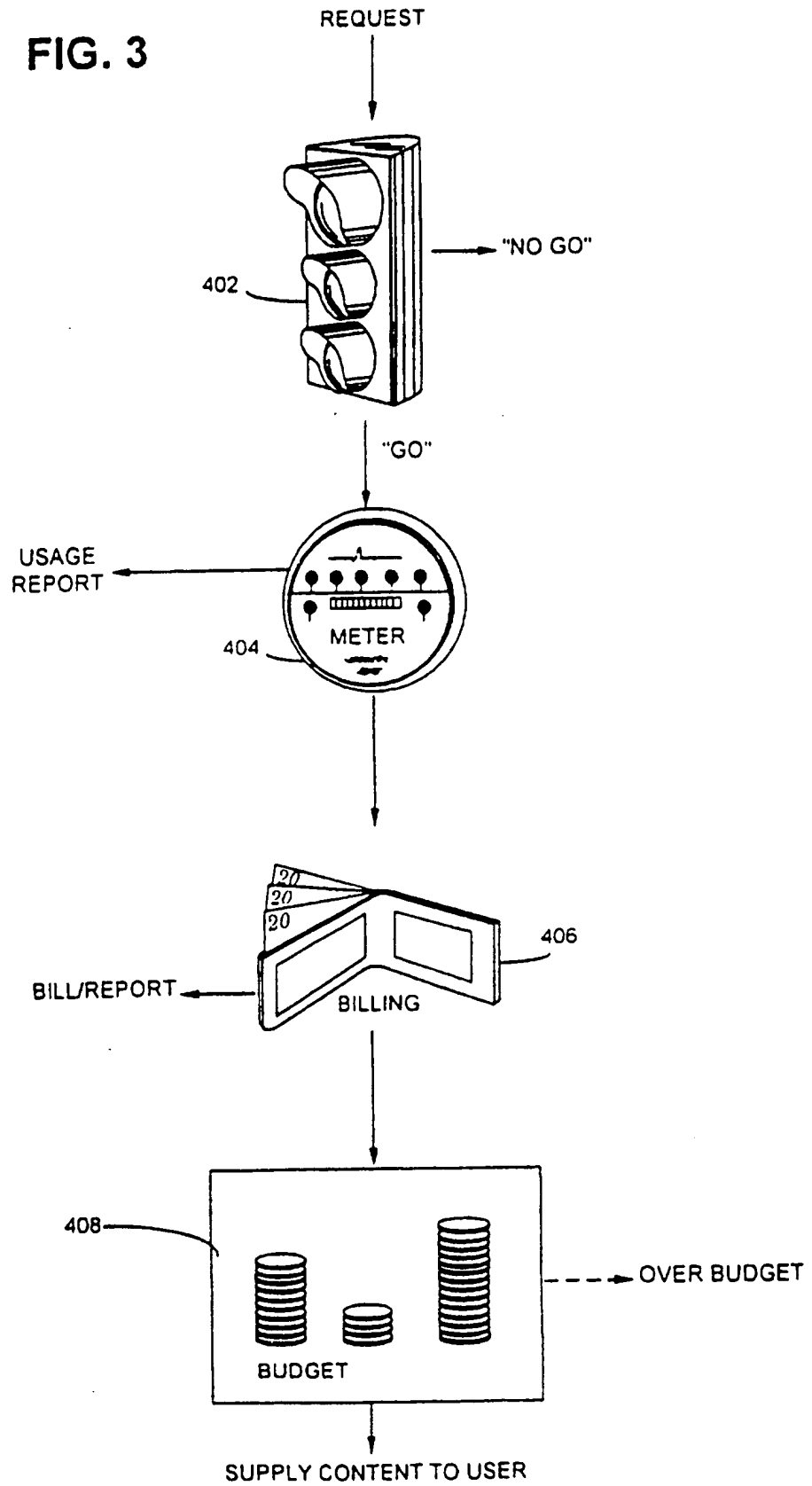
4/163

FIG. 2A



5/163

FIG. 3



6/163

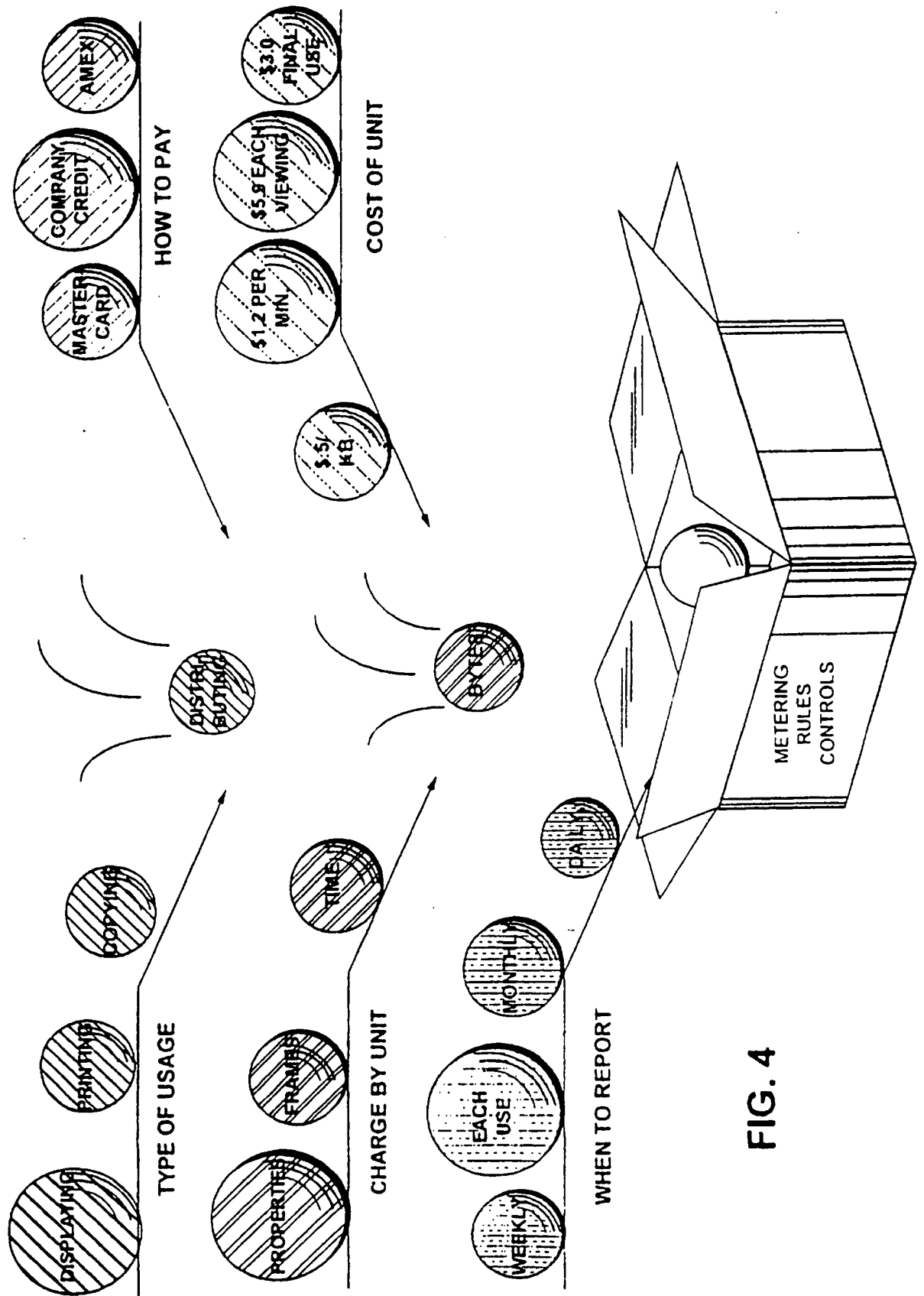
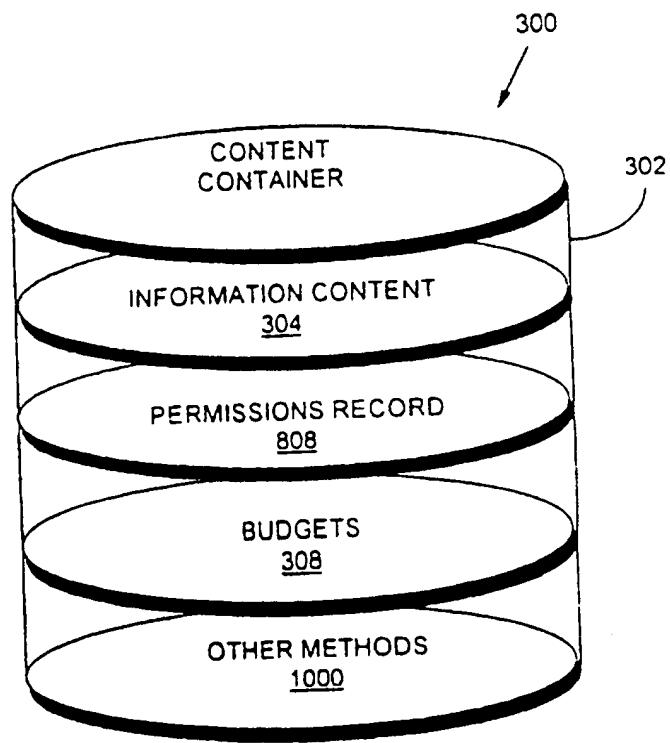


FIG. 4



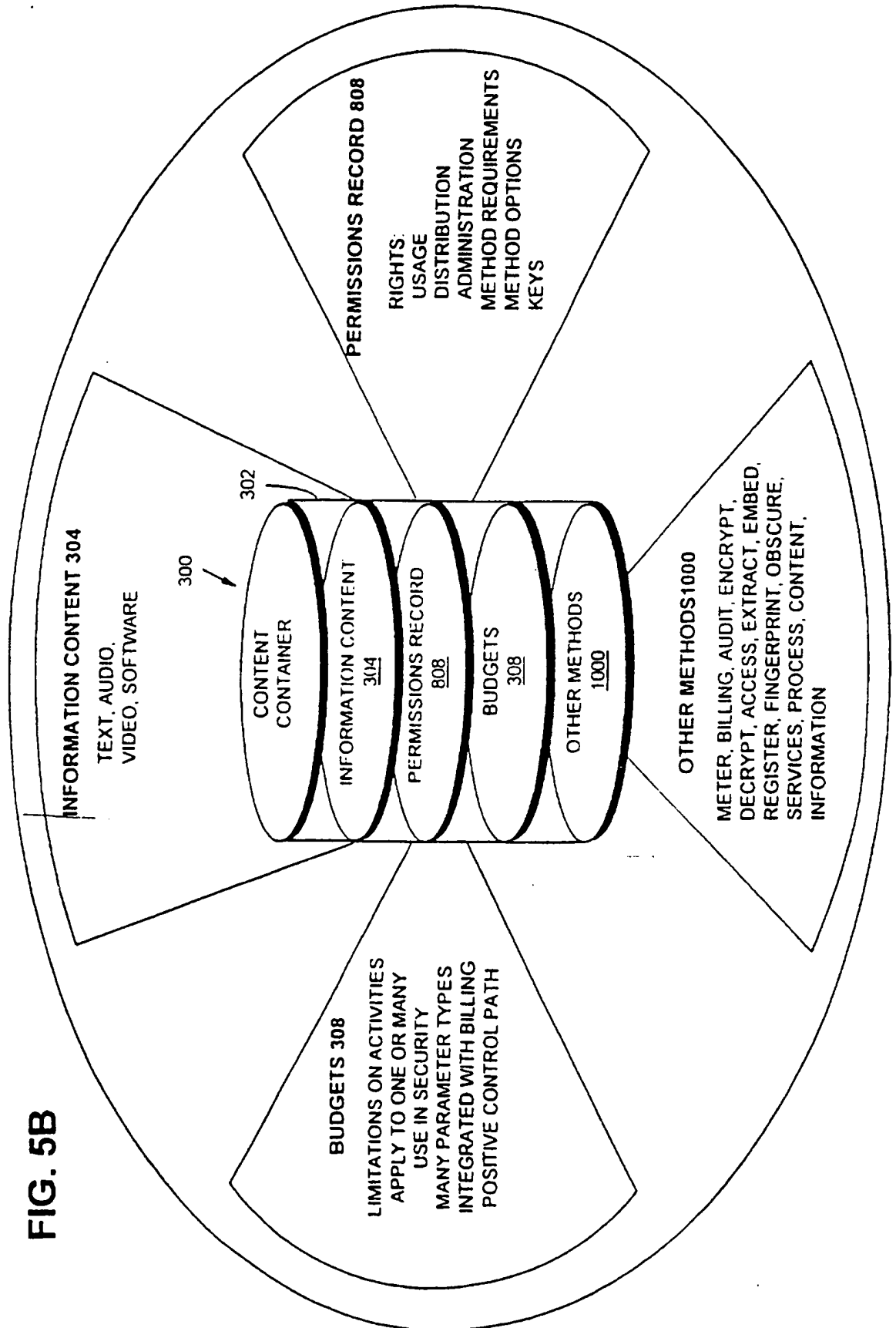
7/163

FIG. 5A



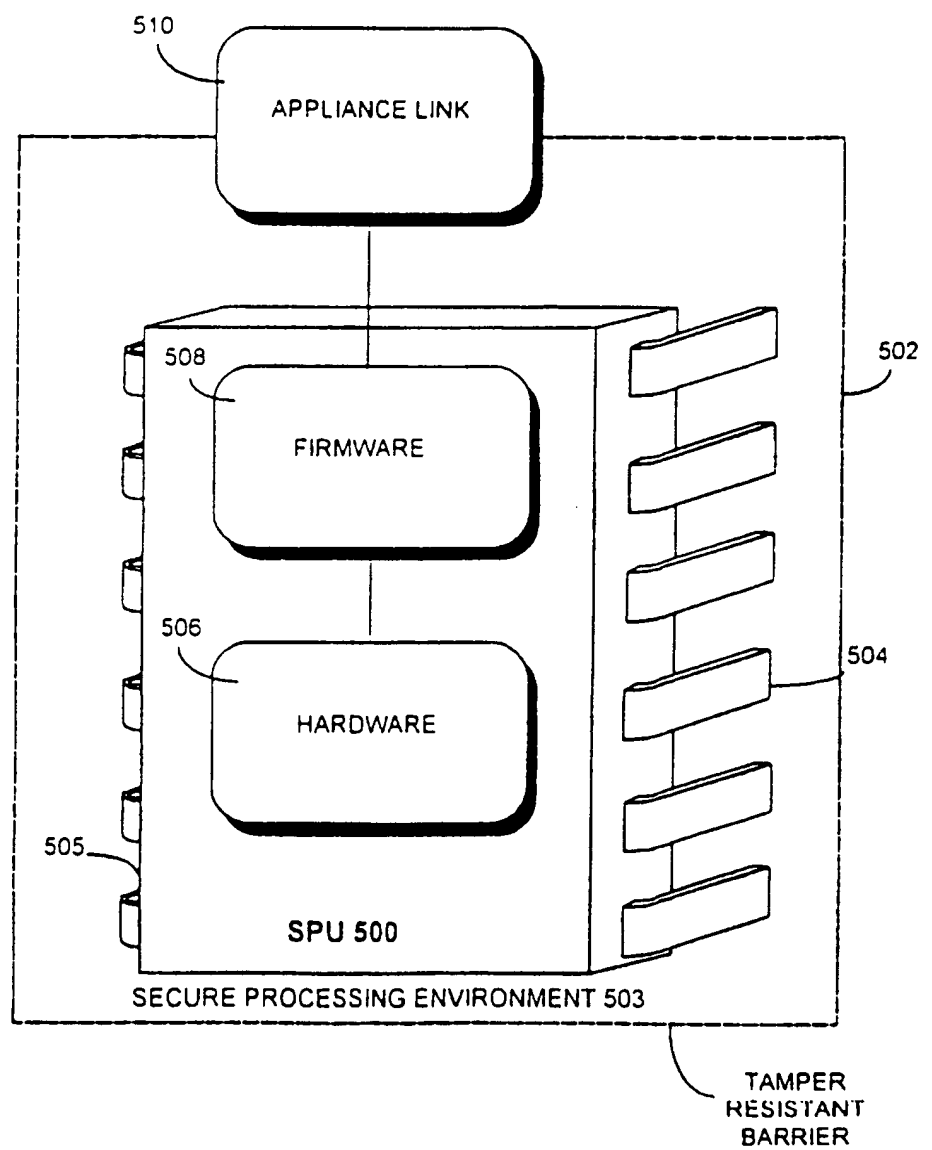
8/163

FIG. 5B

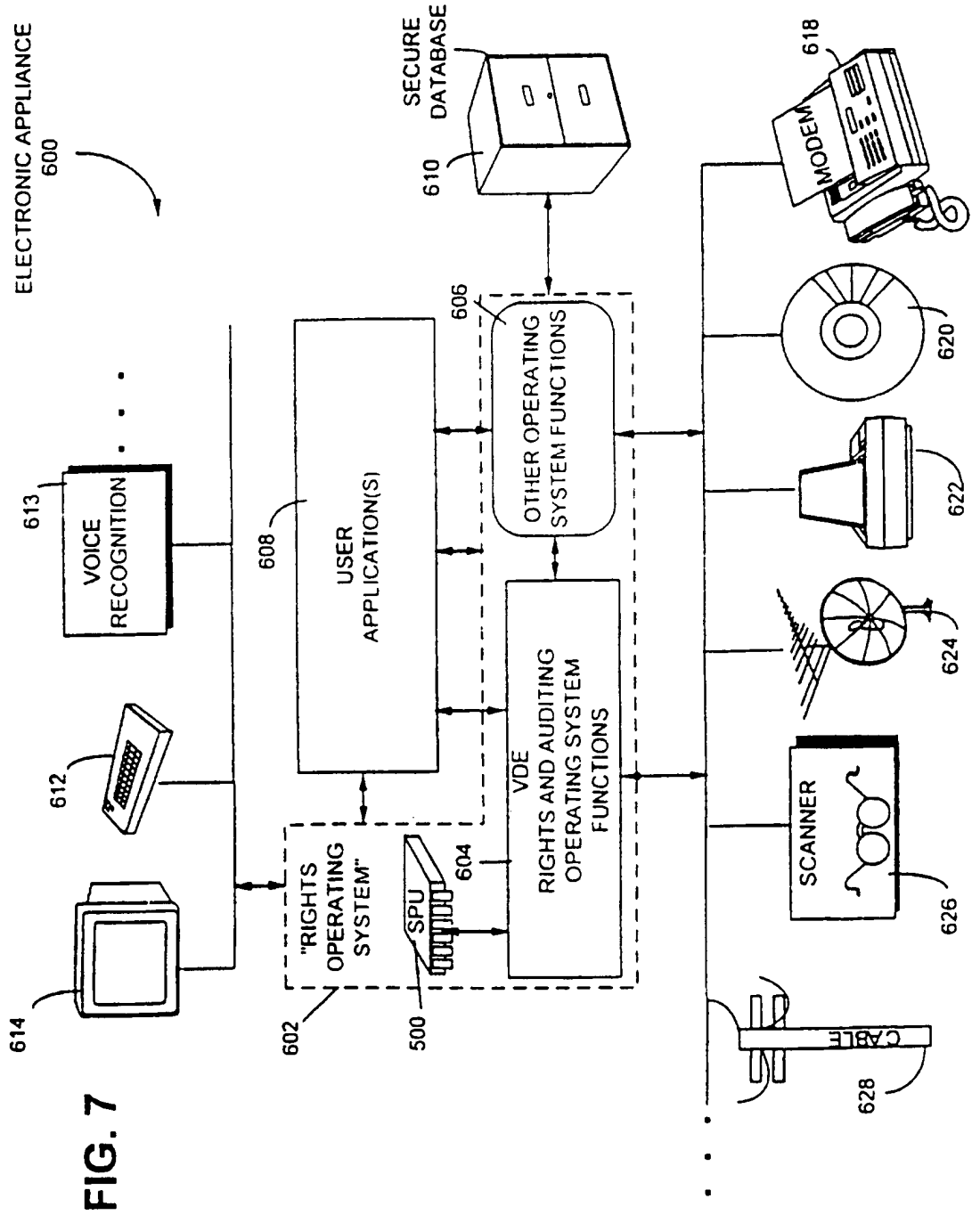


9/163

**FIG. 6**

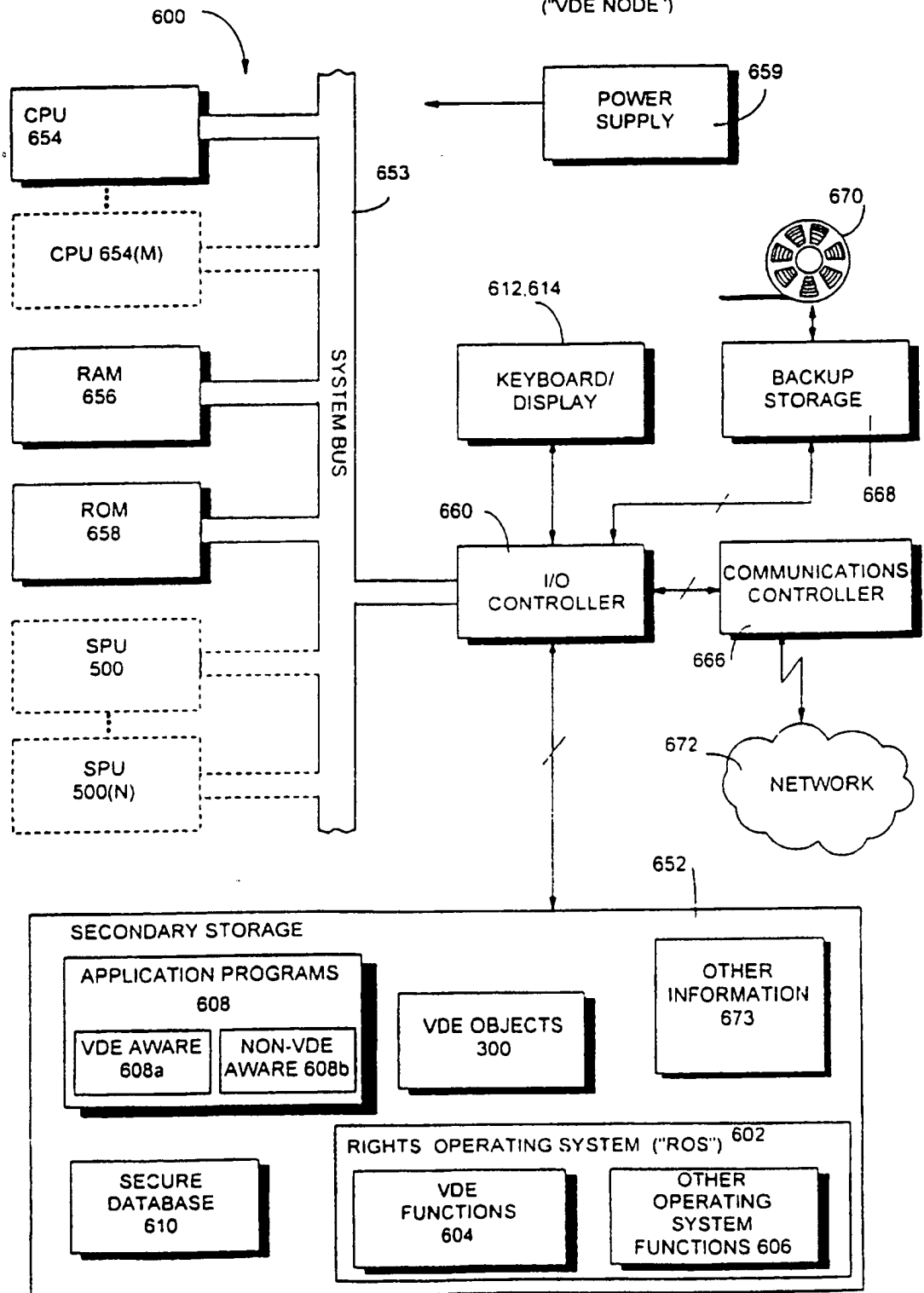


10/163



11/163

**FIG. 8** ELECTRONIC APPLIANCE 600  
("VDE NODE")



12/163

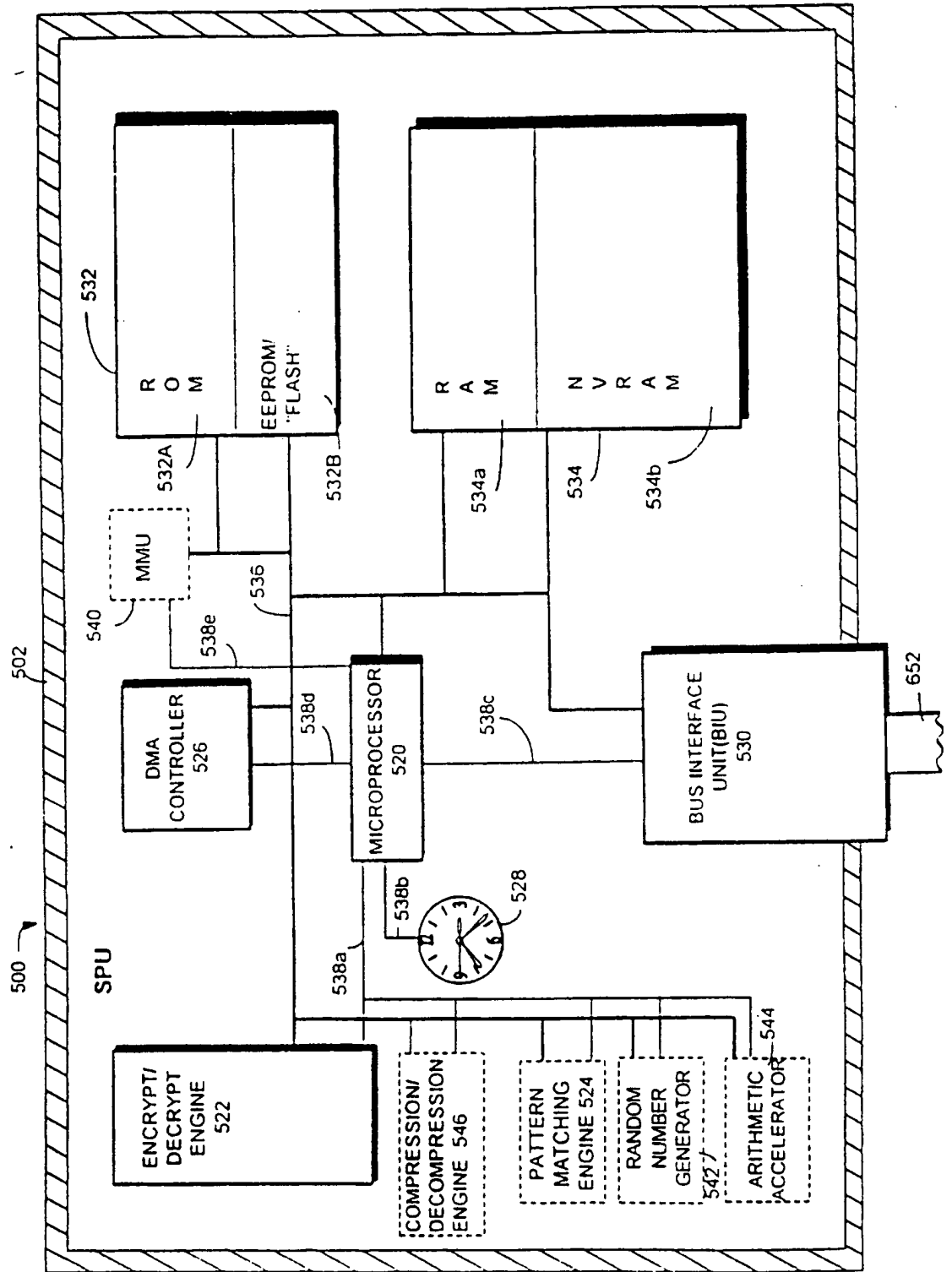


FIG. 9

13/163

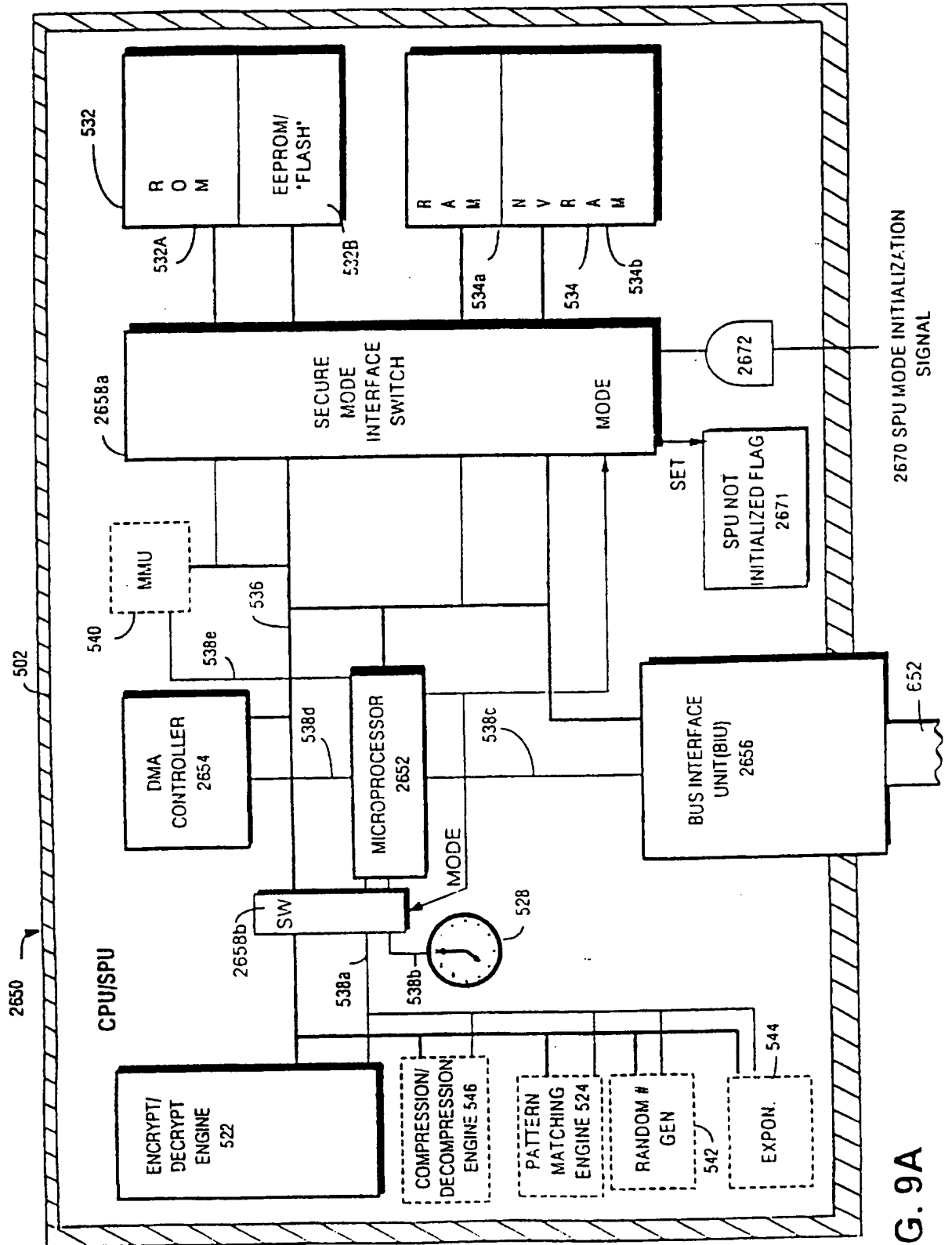


FIG. 9A





15/163

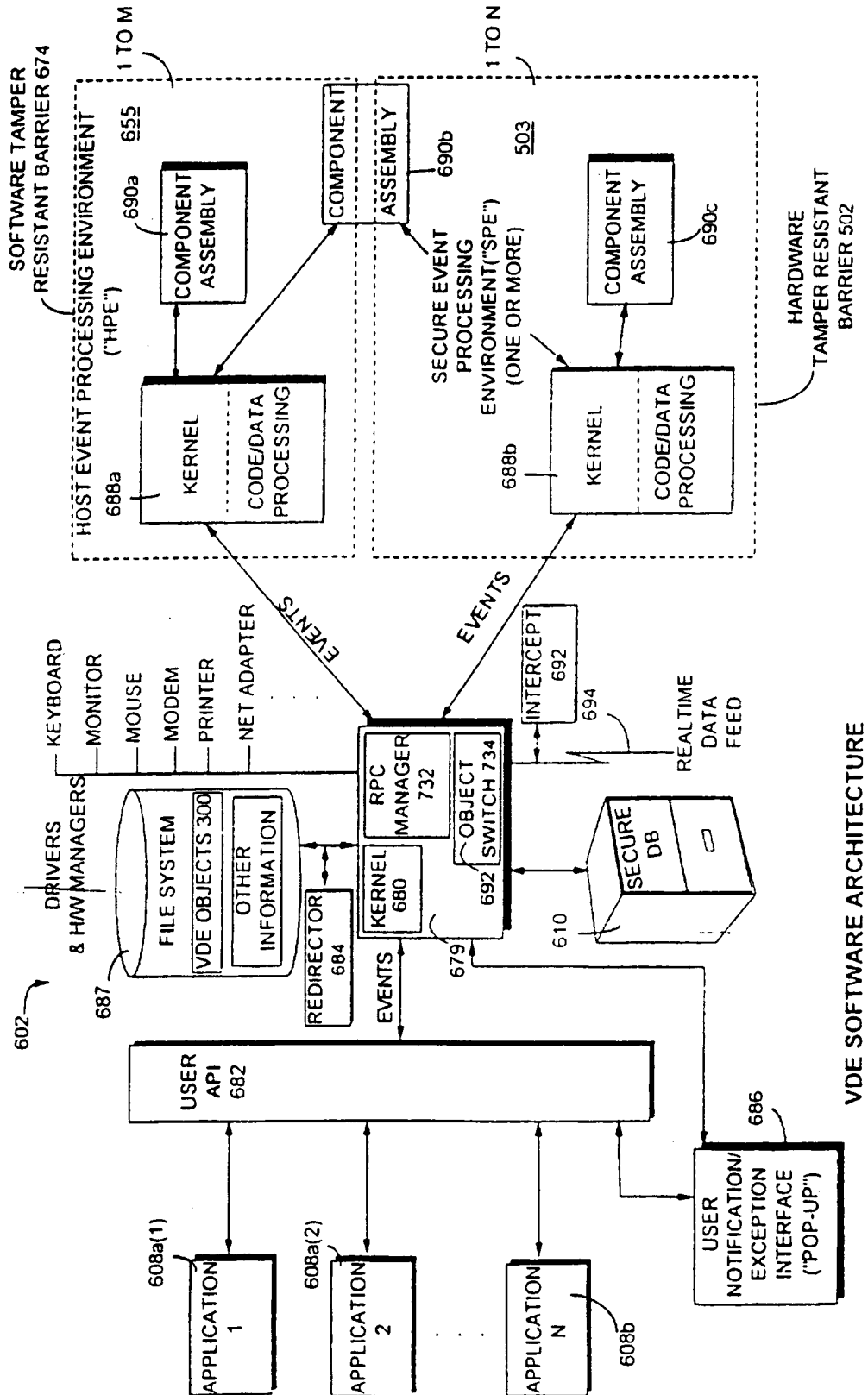
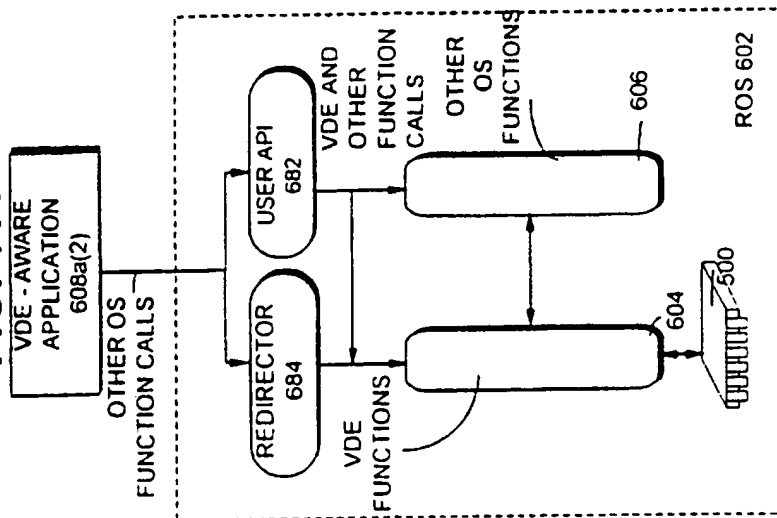
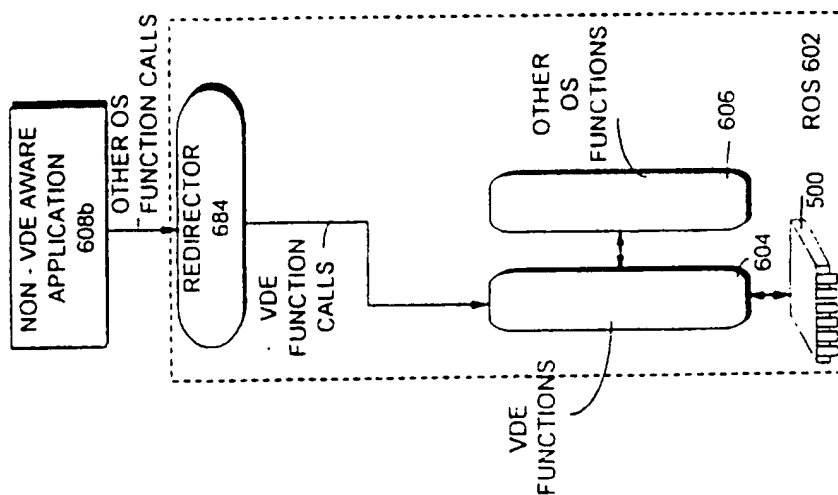


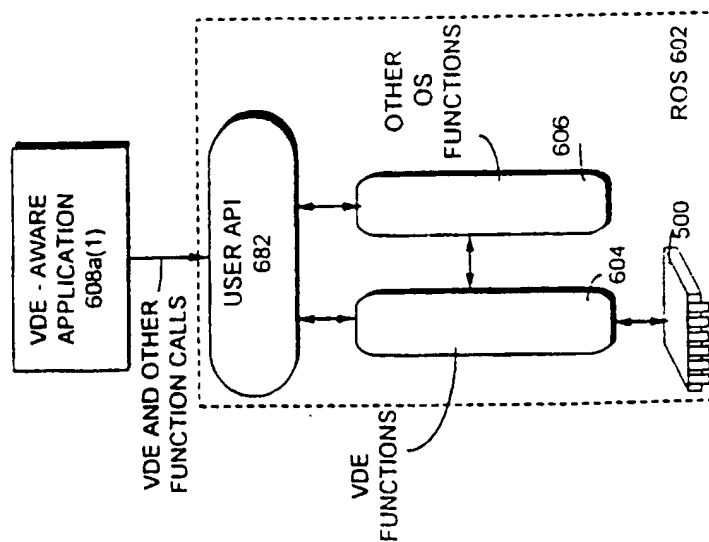
FIG. 10



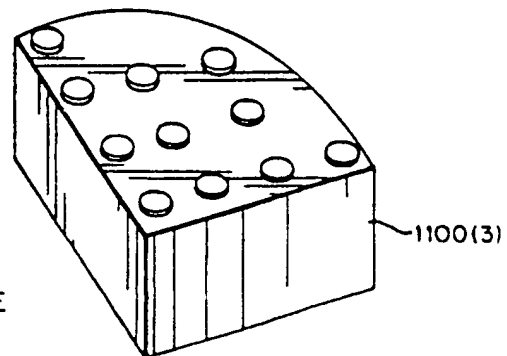
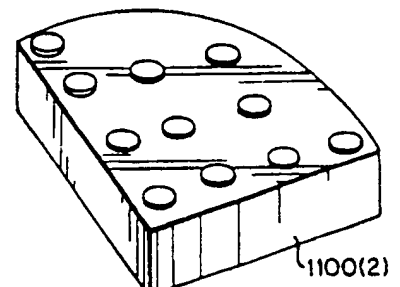
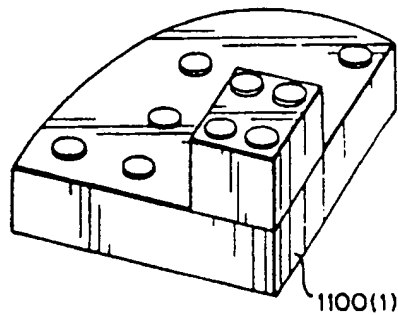
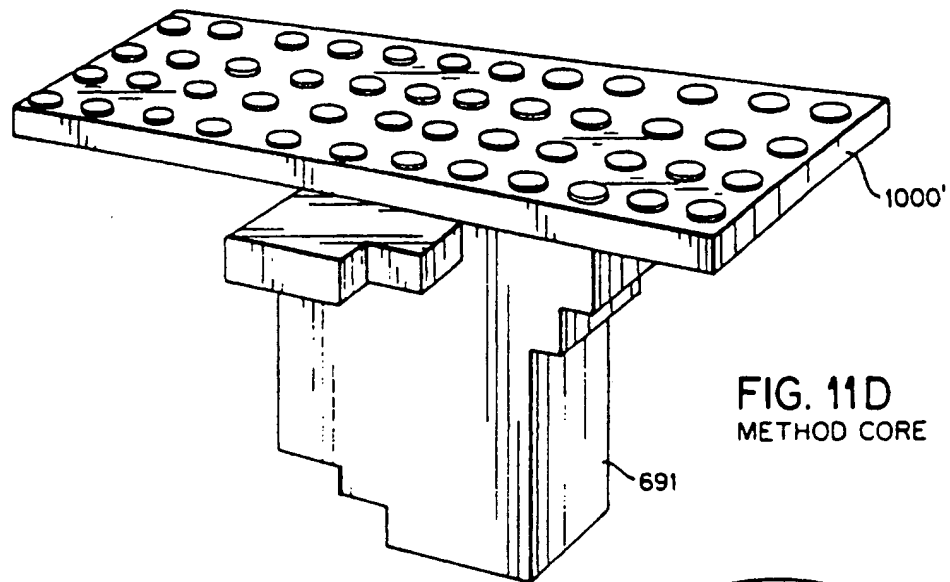
**FIG. 11B**



**FIG. 11A**



17/163



18/163

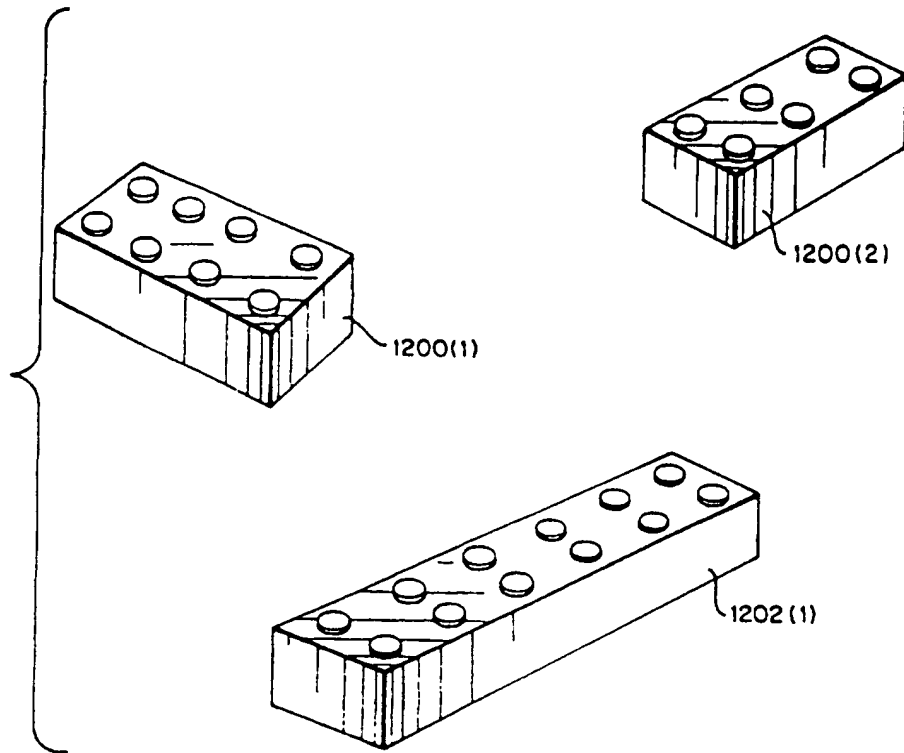
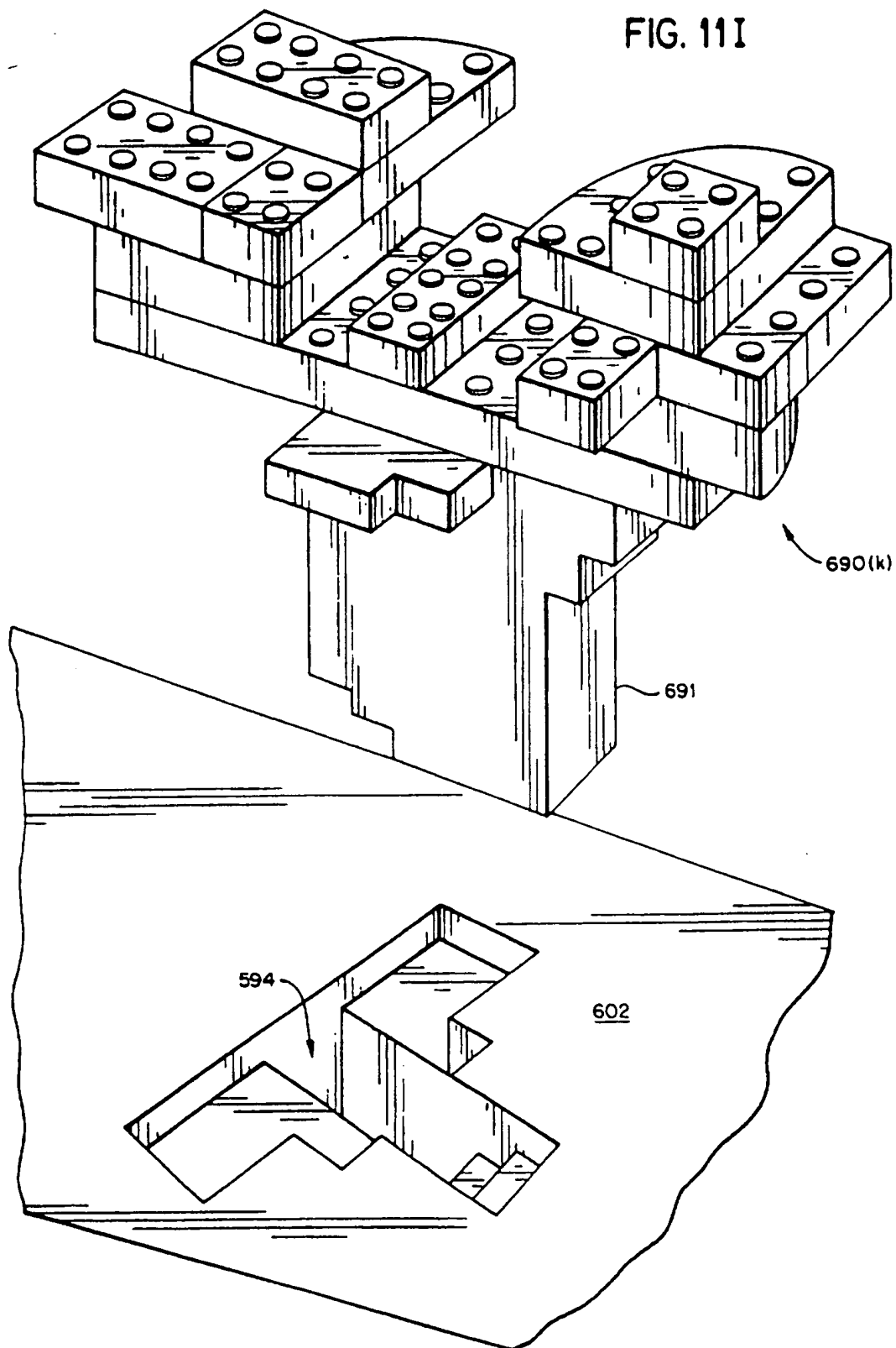


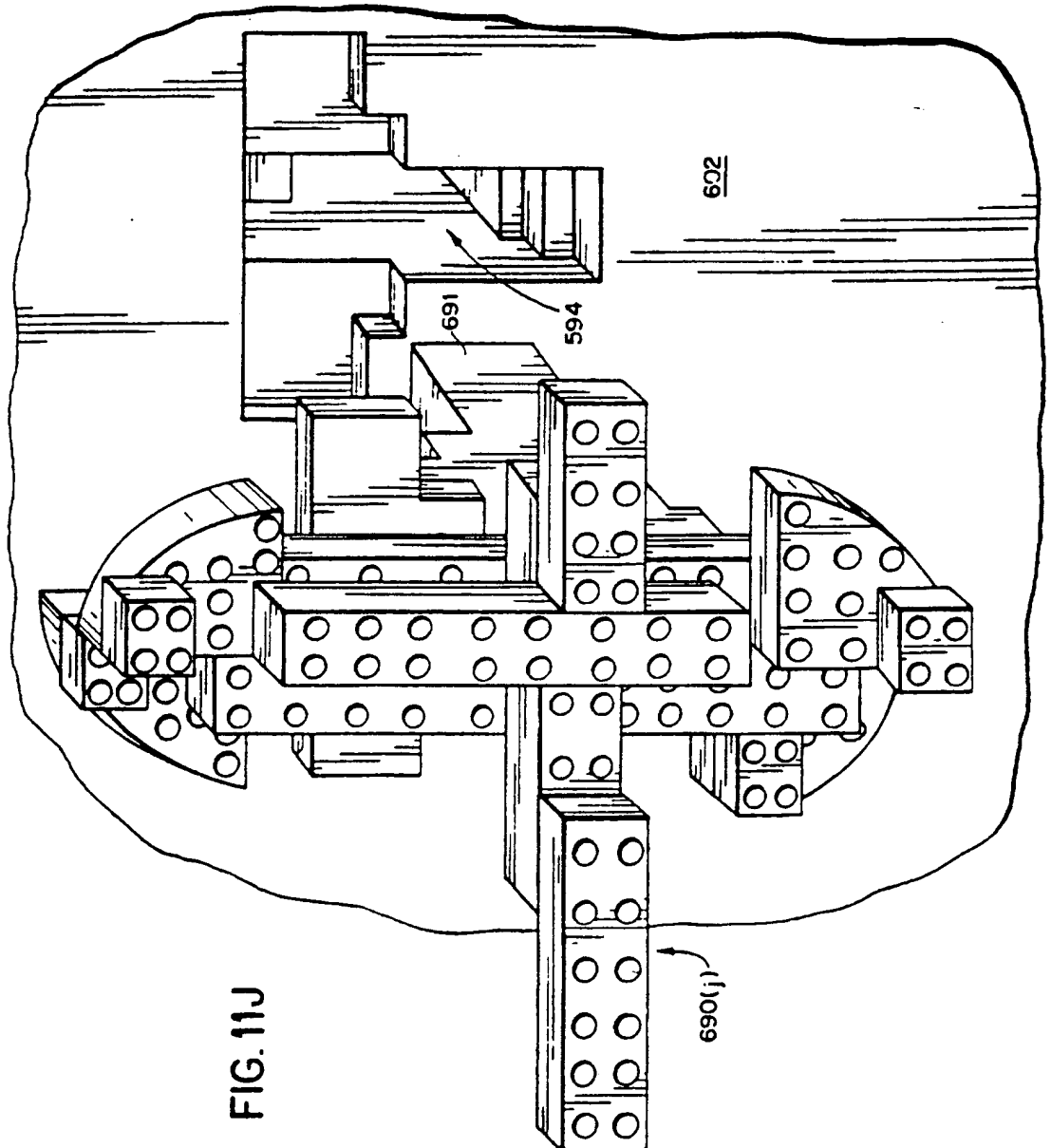
FIG. 11H  
DATA STRUCTURES

19/163

FIG. 11I



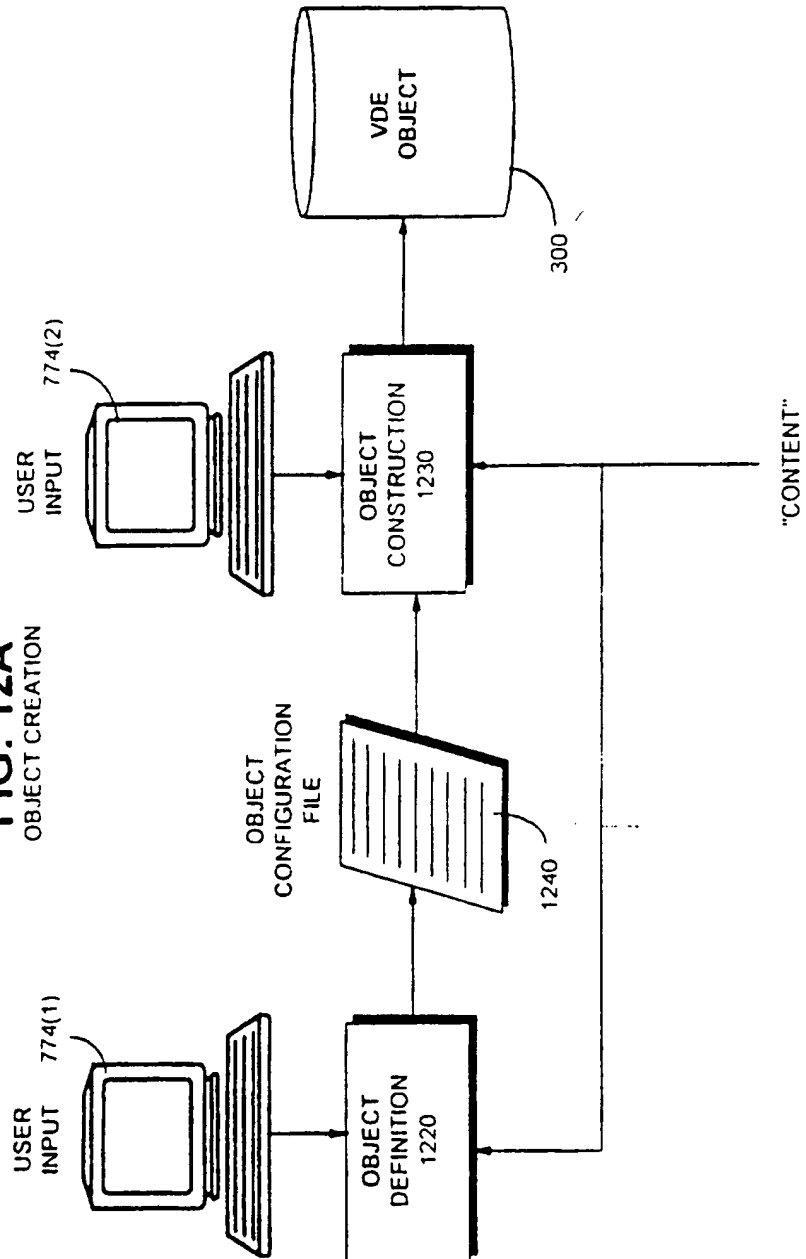
20/163





22/163

**FIG. 12A**  
OBJECT CREATION

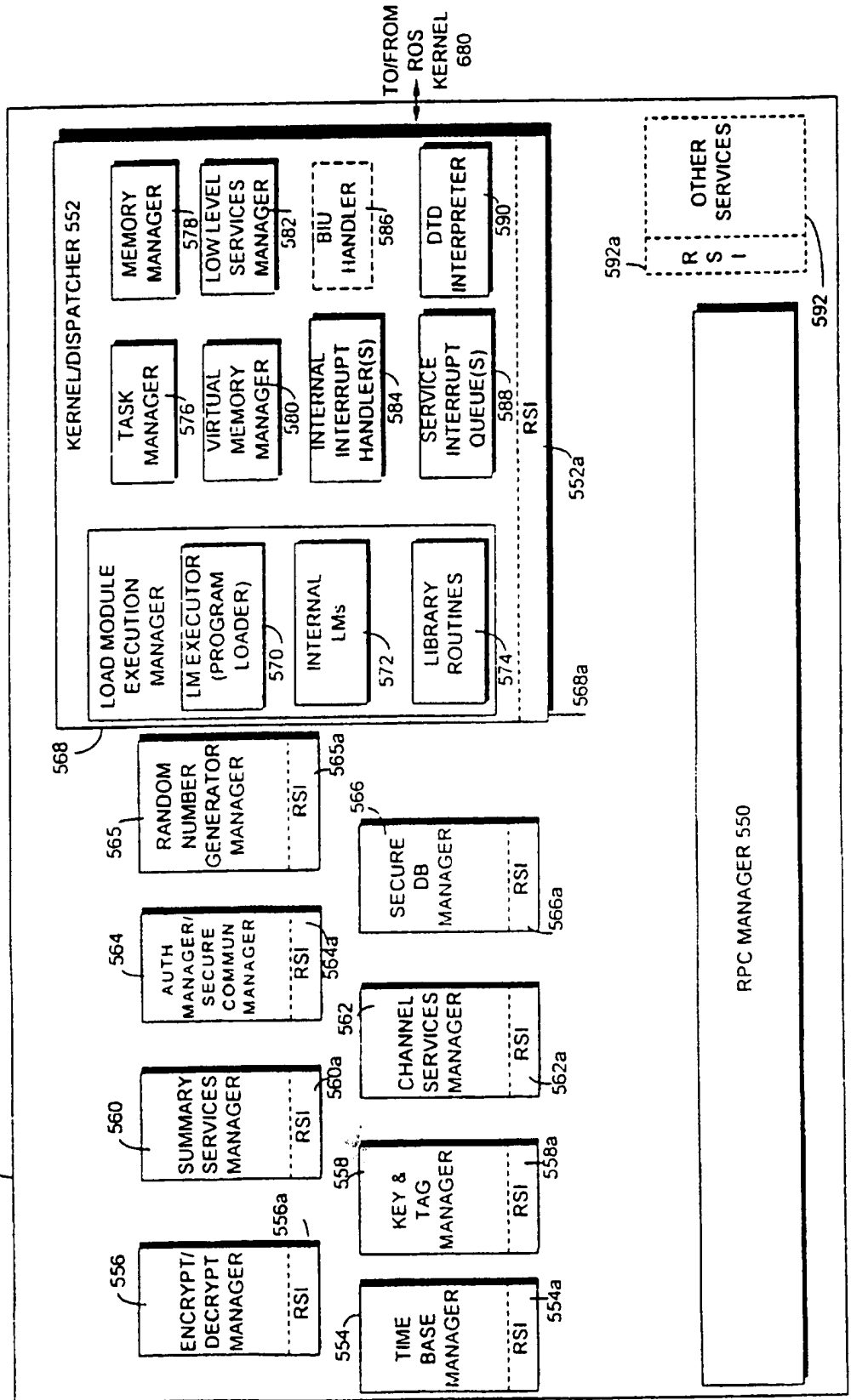




23/163

FIG. 13

PROTECTED PROCESSING ENVIRONMENT 650



24/163

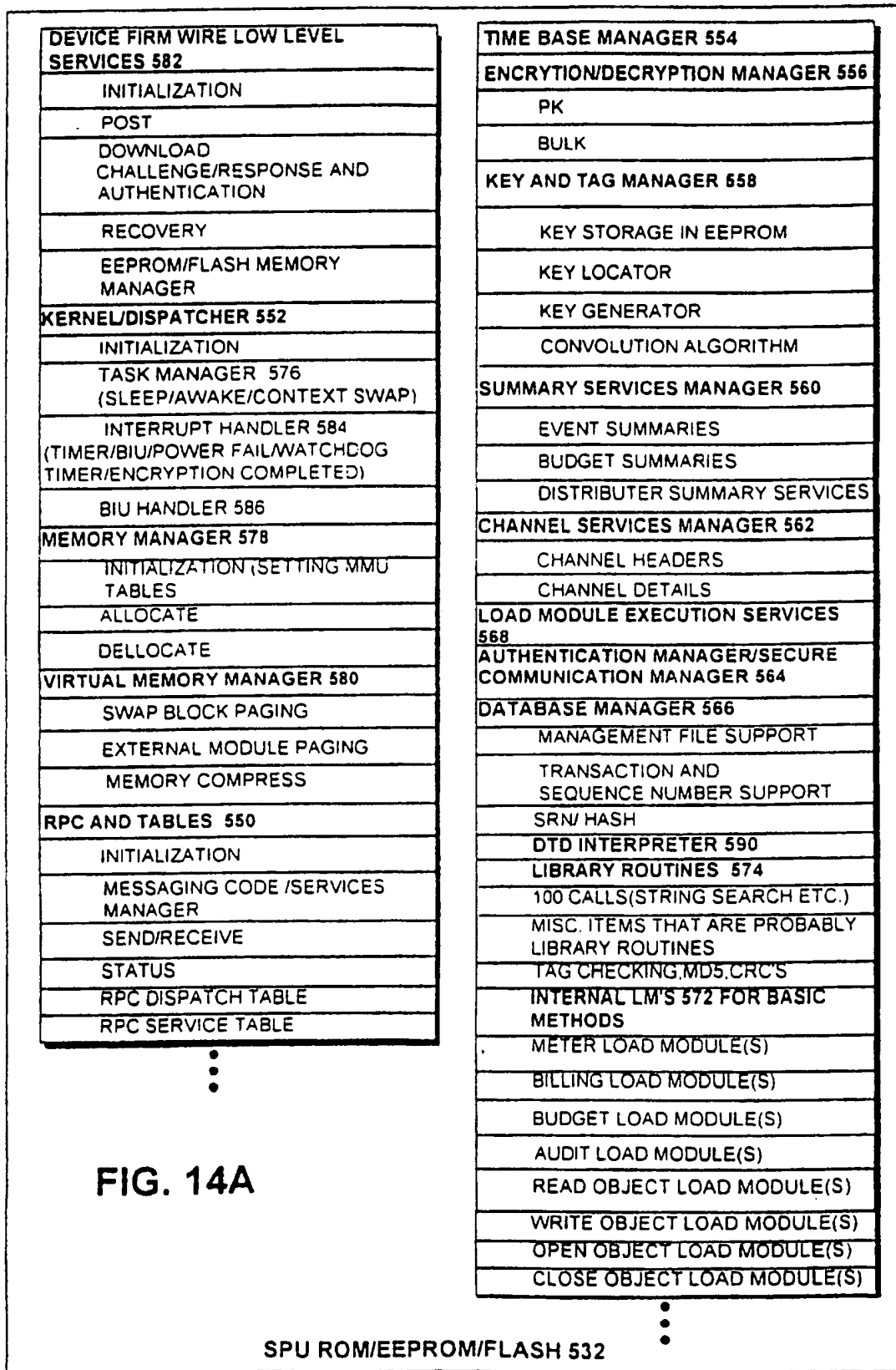
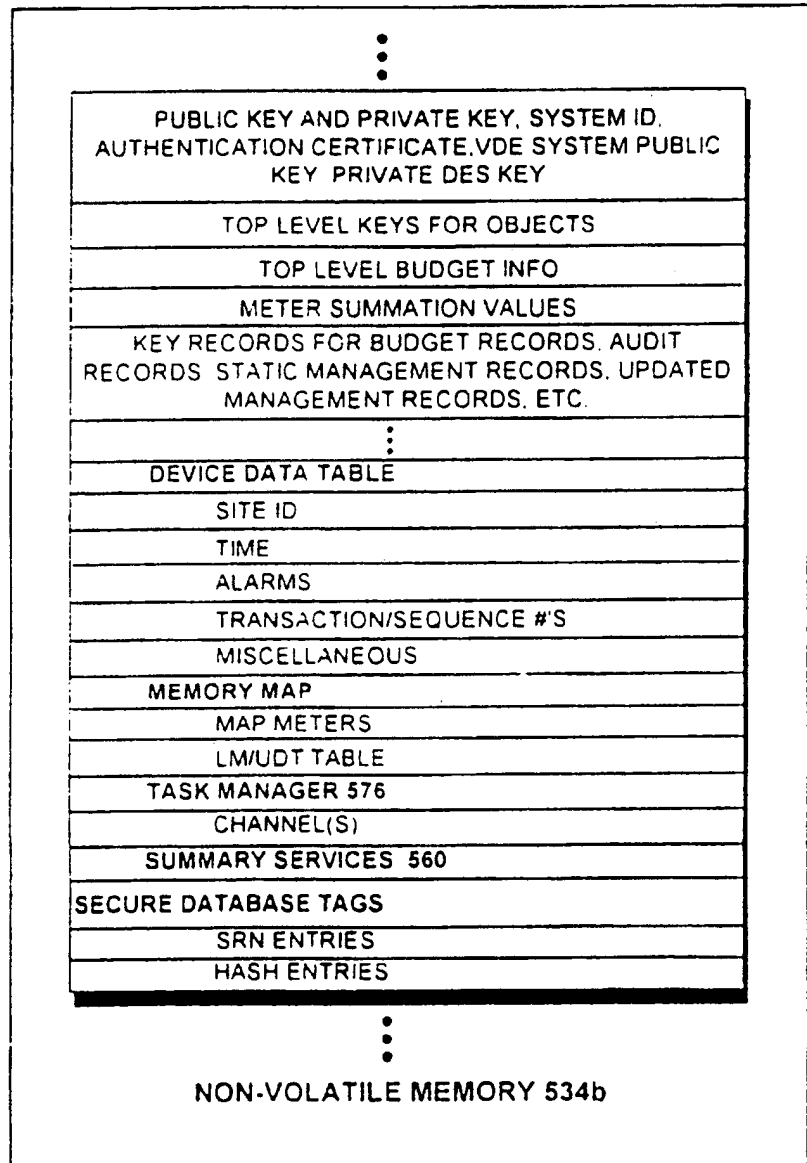


FIG. 14A

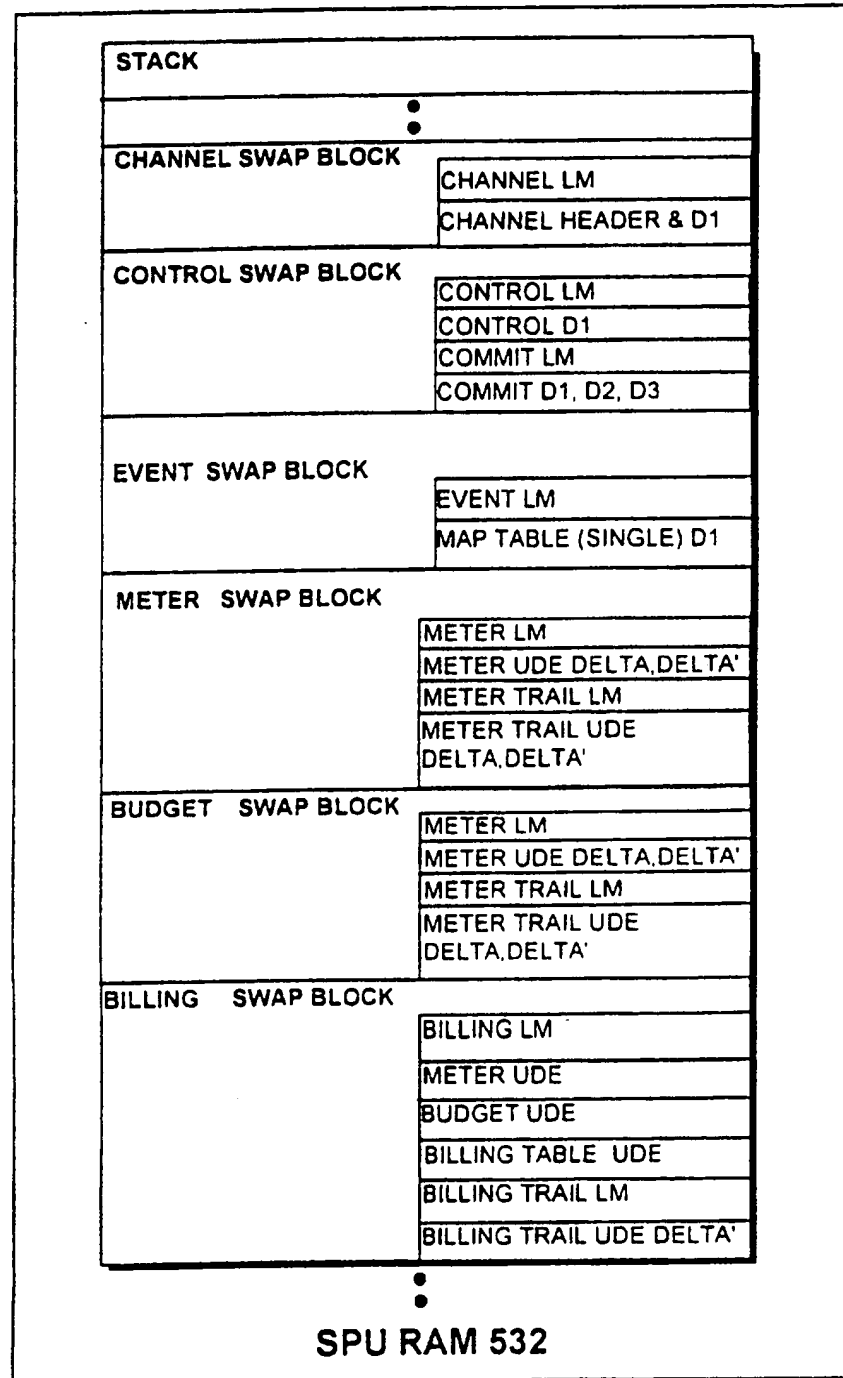
25/163

FIG. 14B



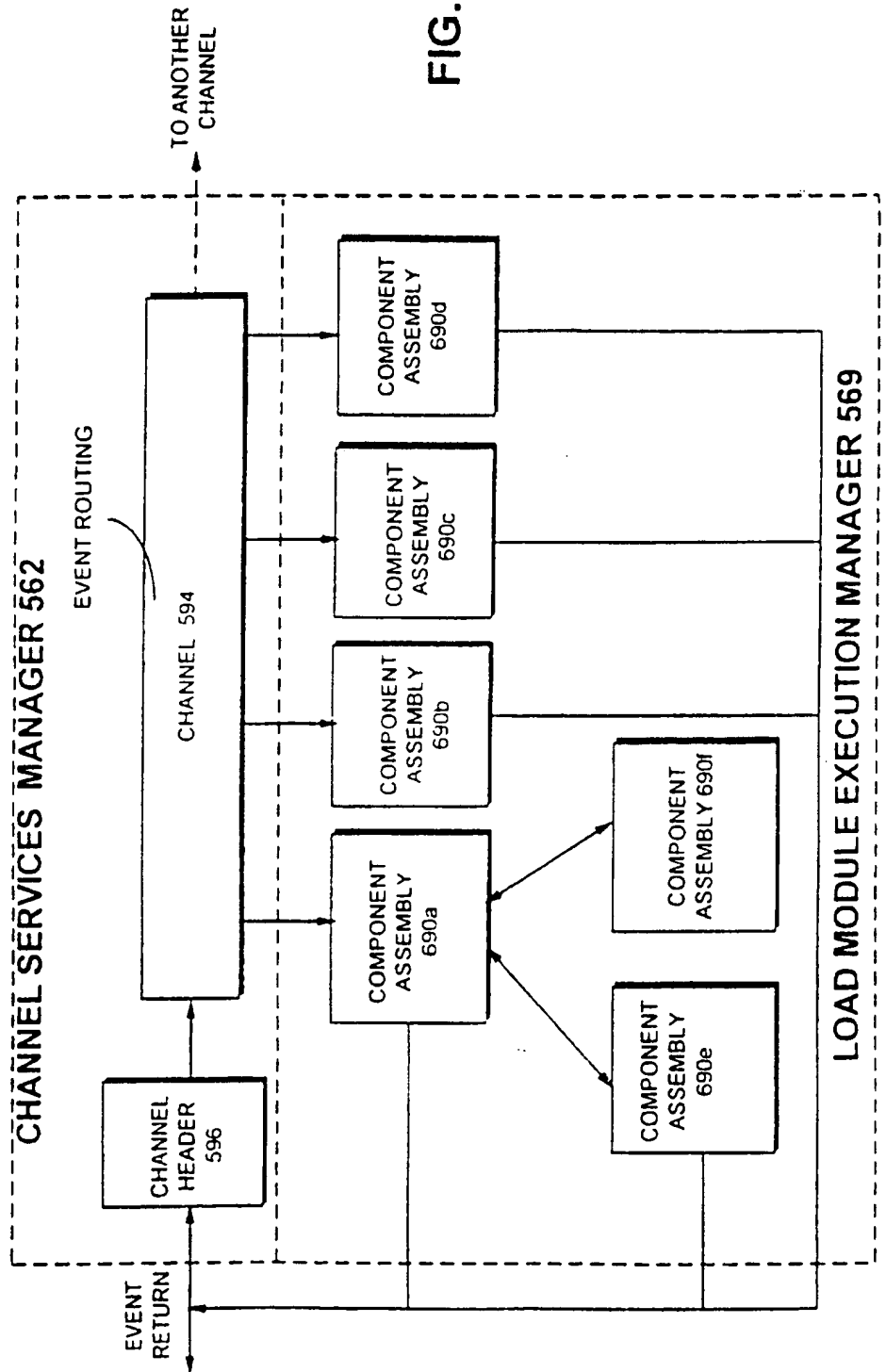
26/163

FIG. 14C



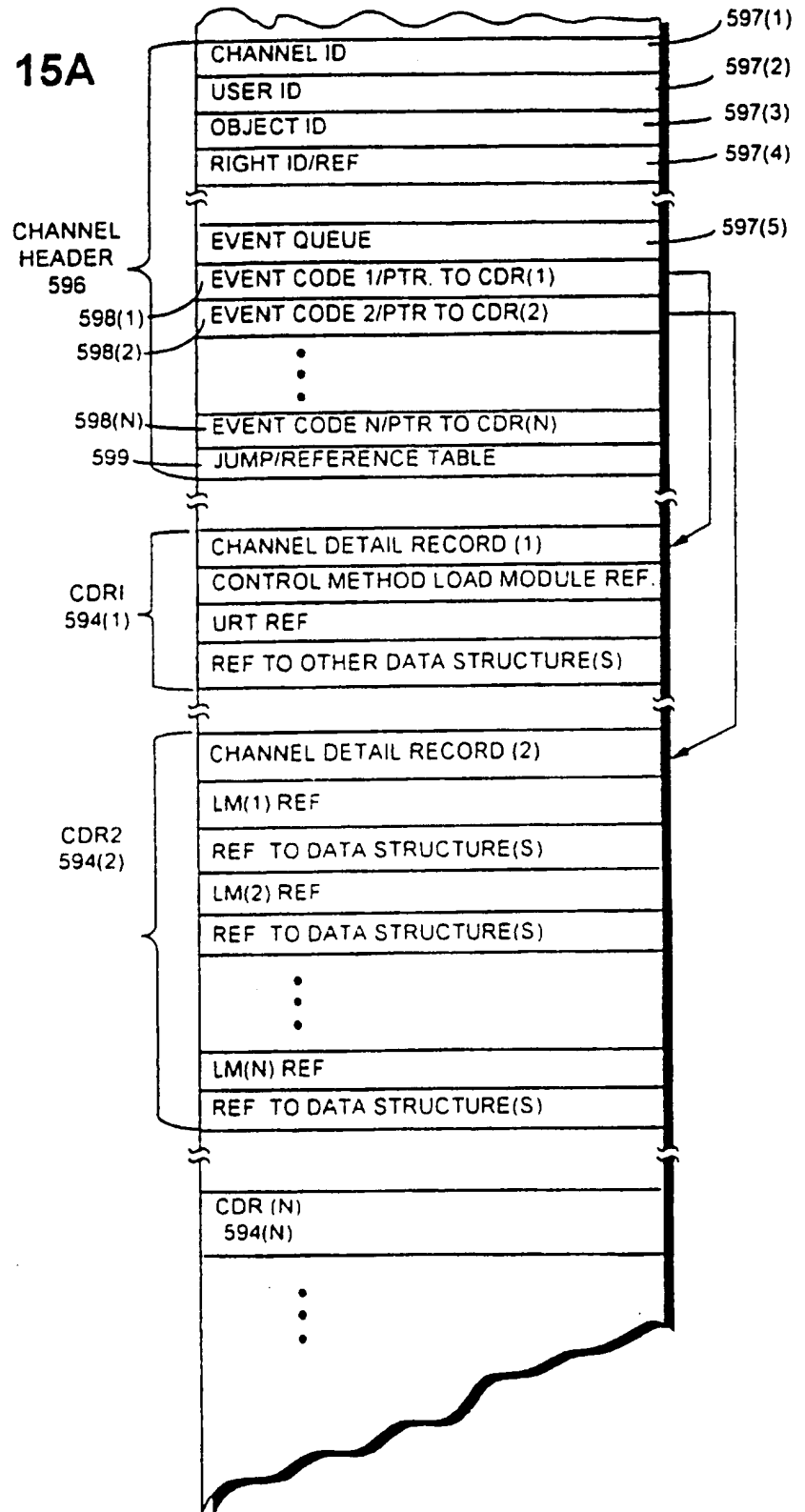
27/163

FIG. 15



28/163

FIG. 15A



29/163

FIG. 15B

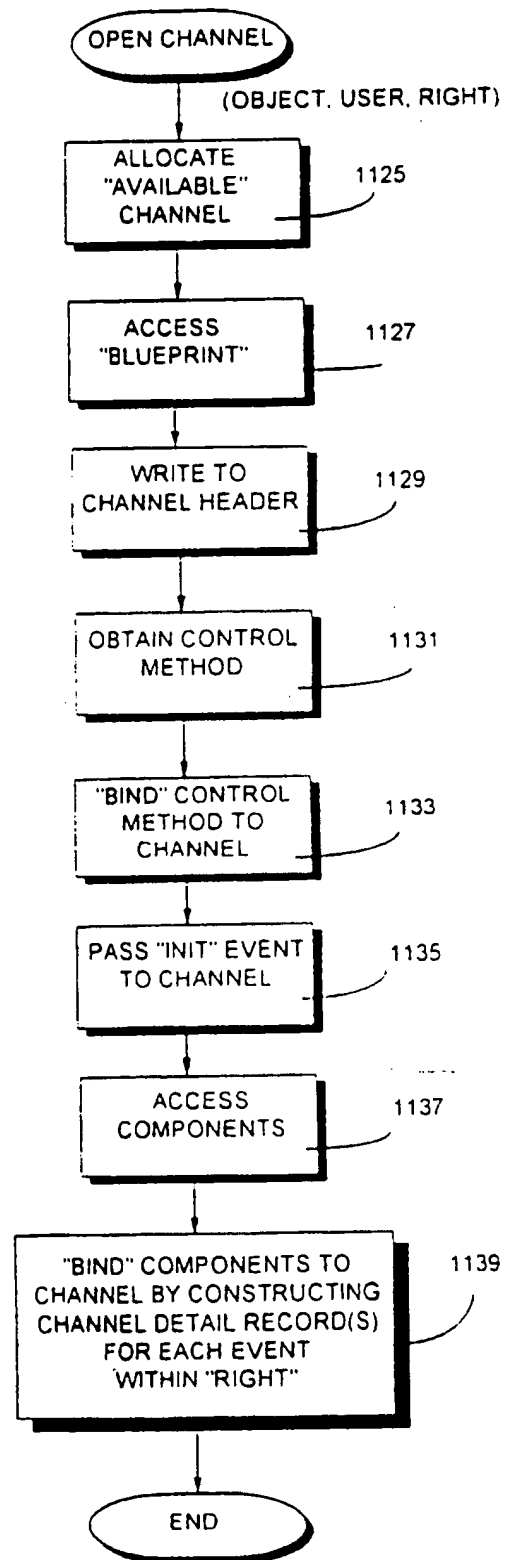
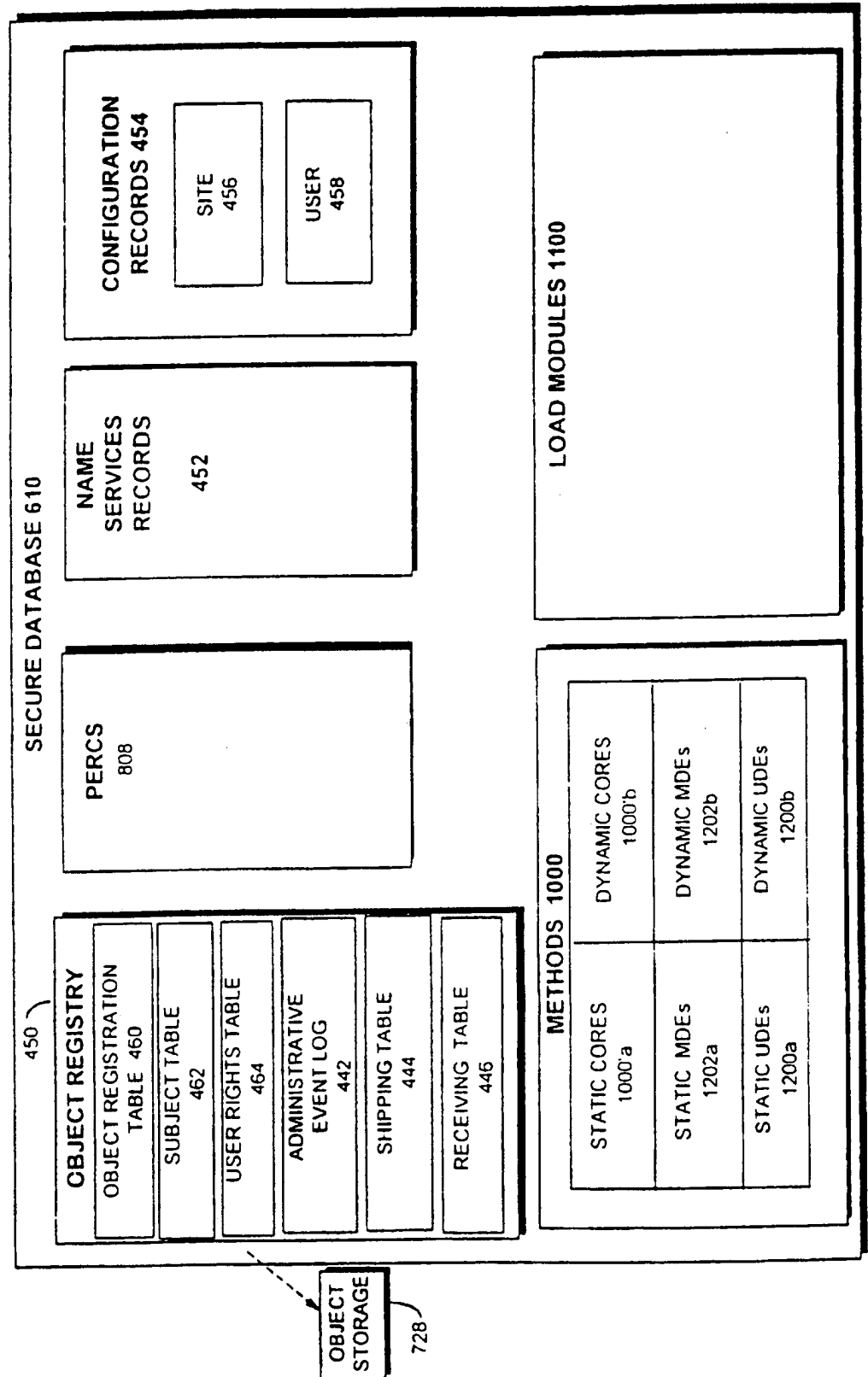
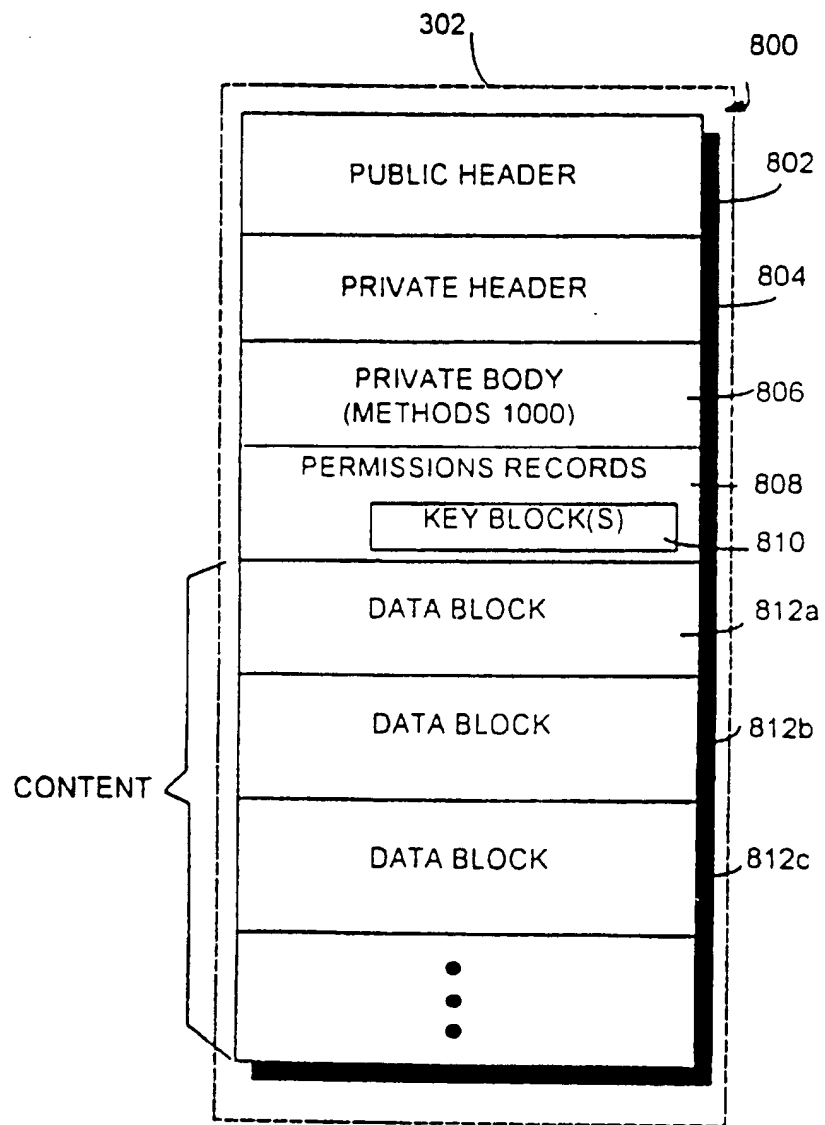


FIG. 16



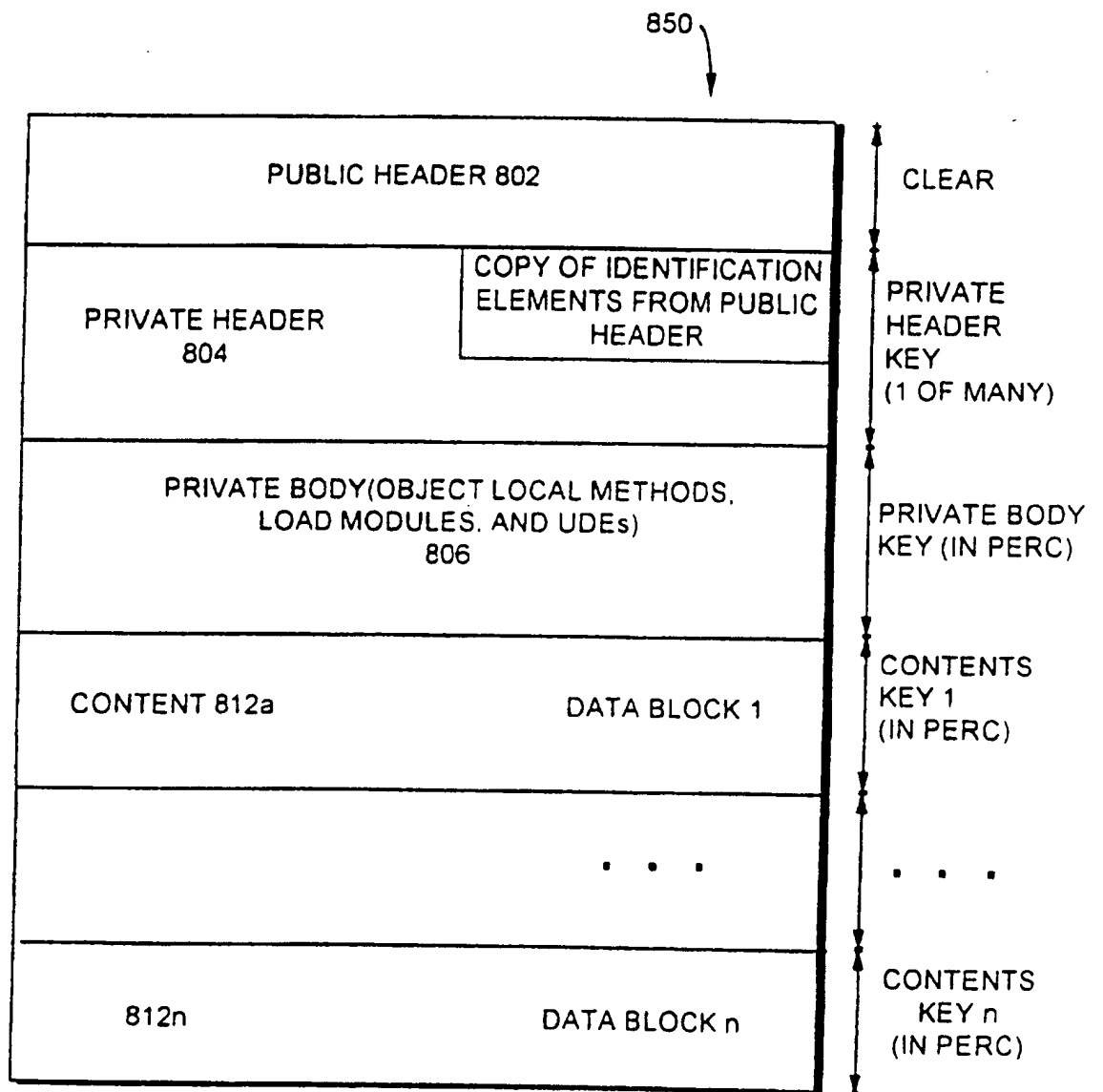




LOGICAL OBJECT

FIG. 17

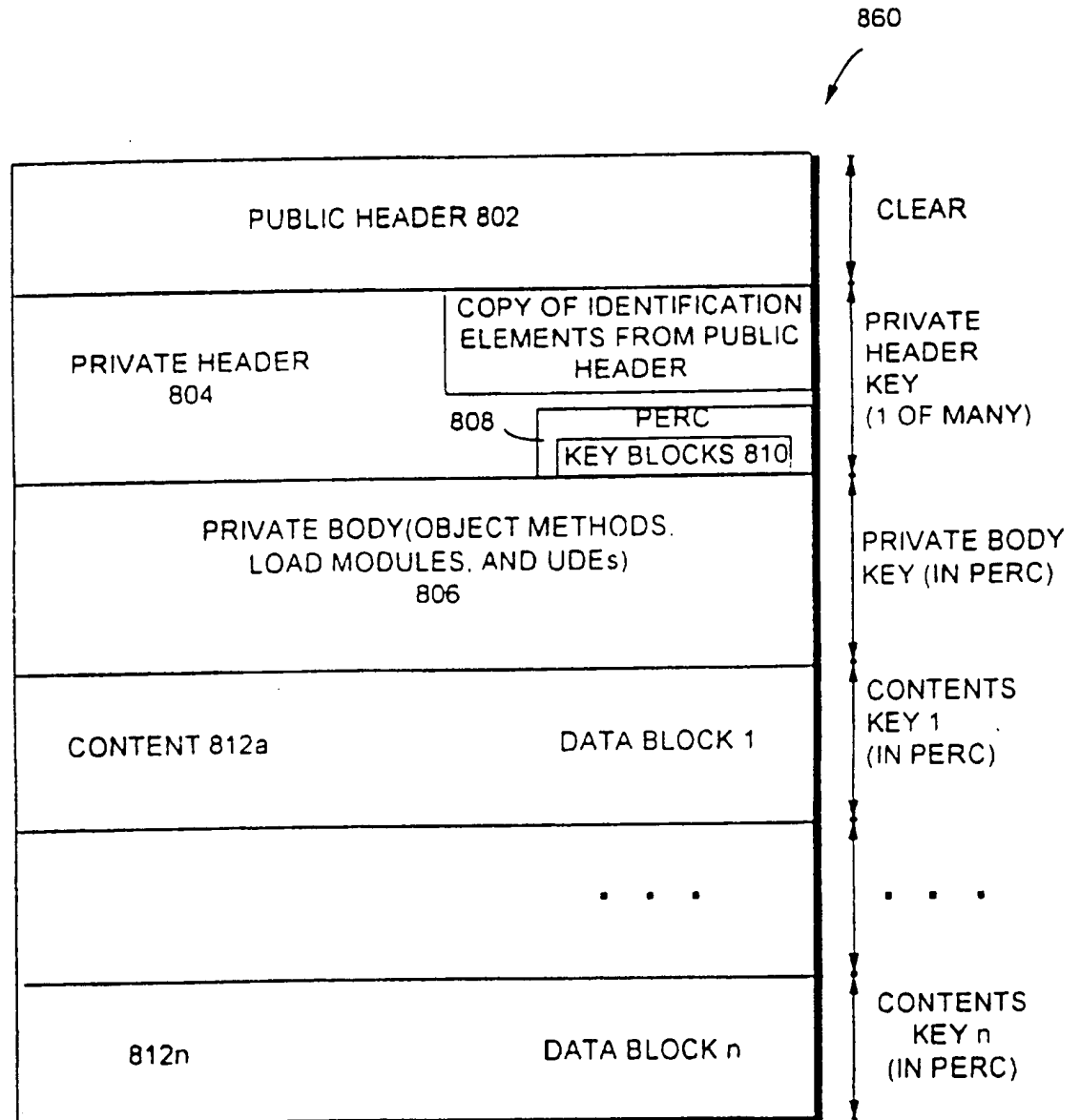
32/163



STATIONARY OBJECT

FIG. 18

33/163

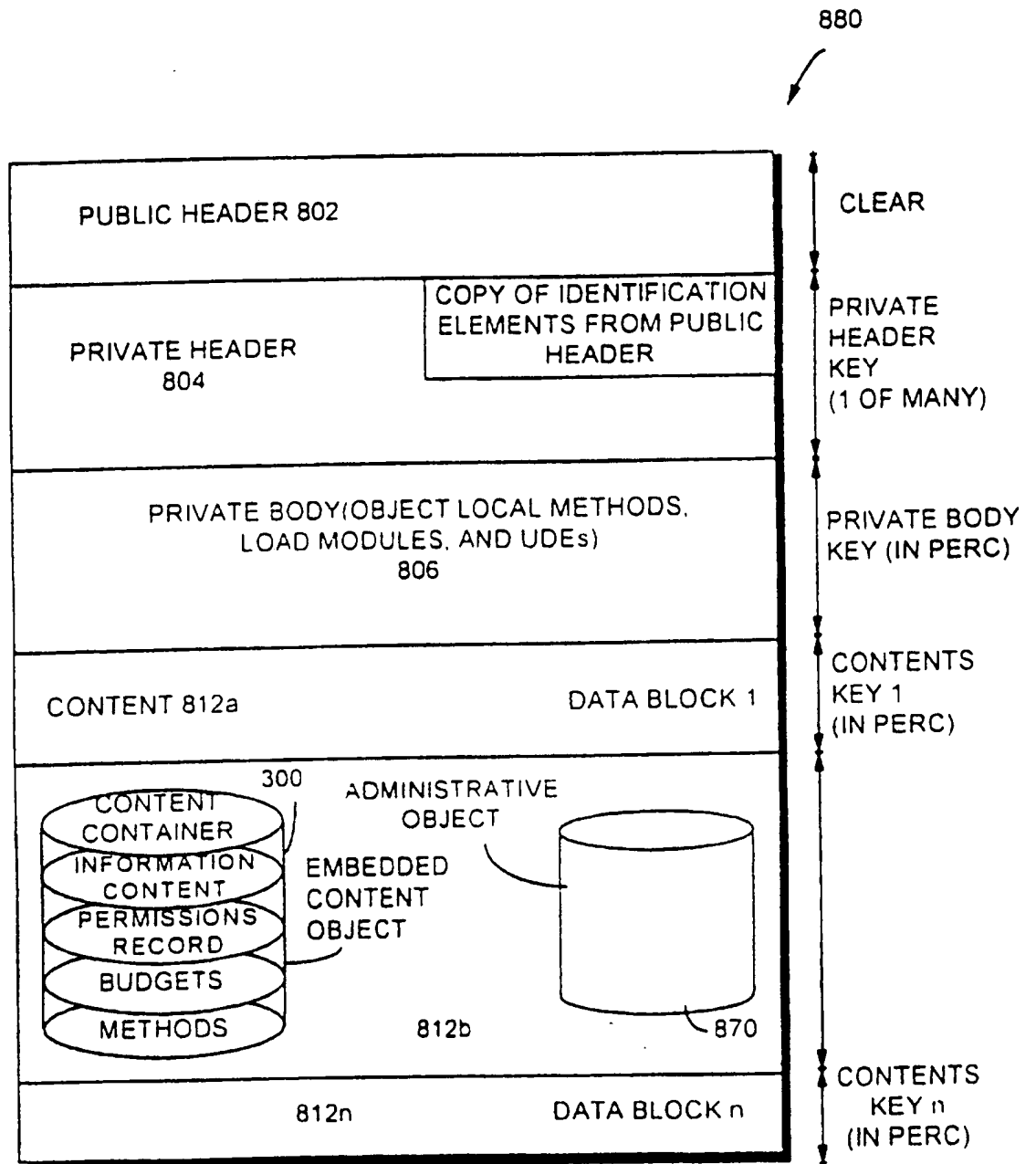


TRAVELING OBJECT

FIG. 19

SUBSTITUTE SHEET (RULE 26)

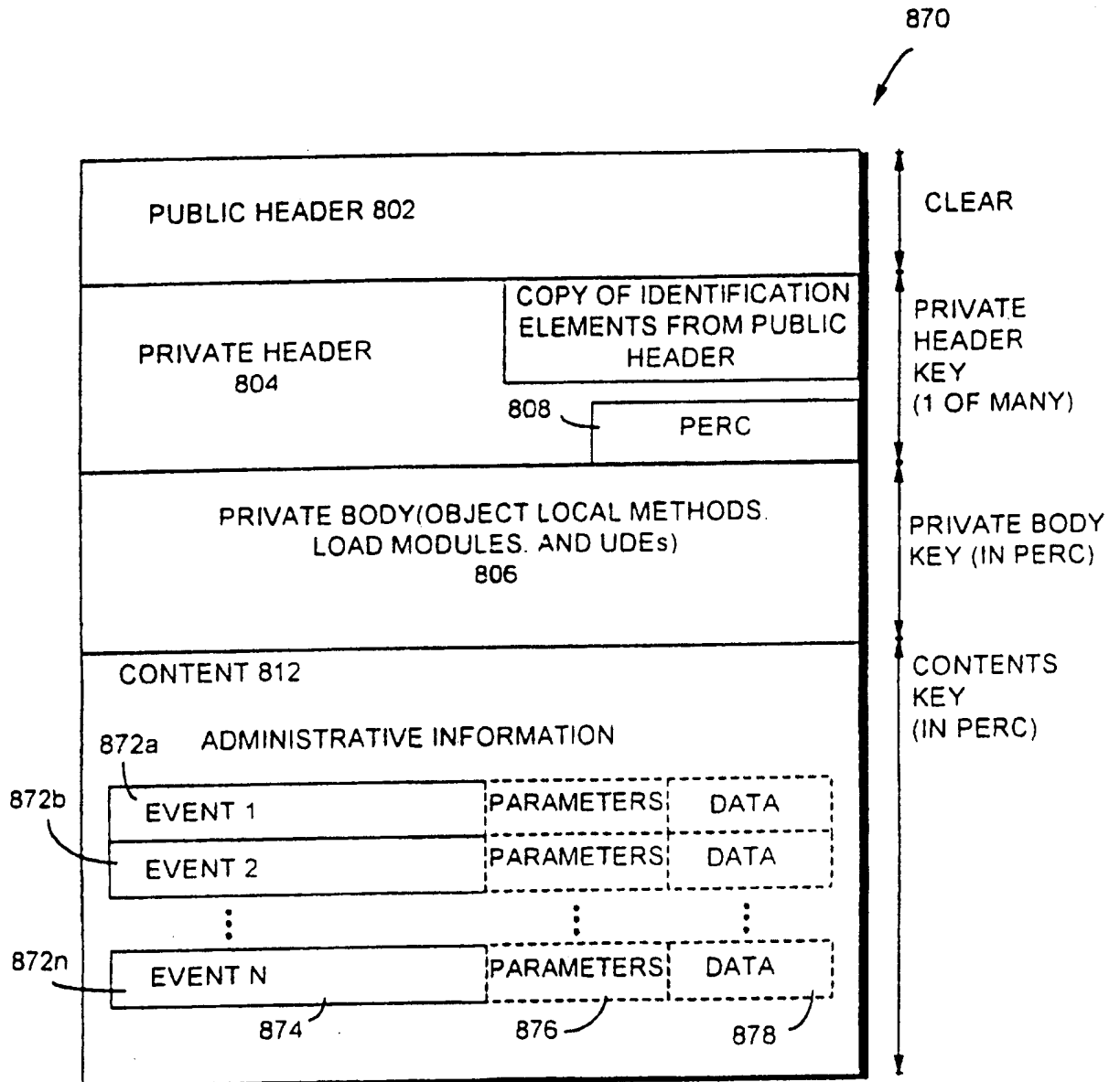
34/163



CONTENT OBJECT

FIG. 20

35/163

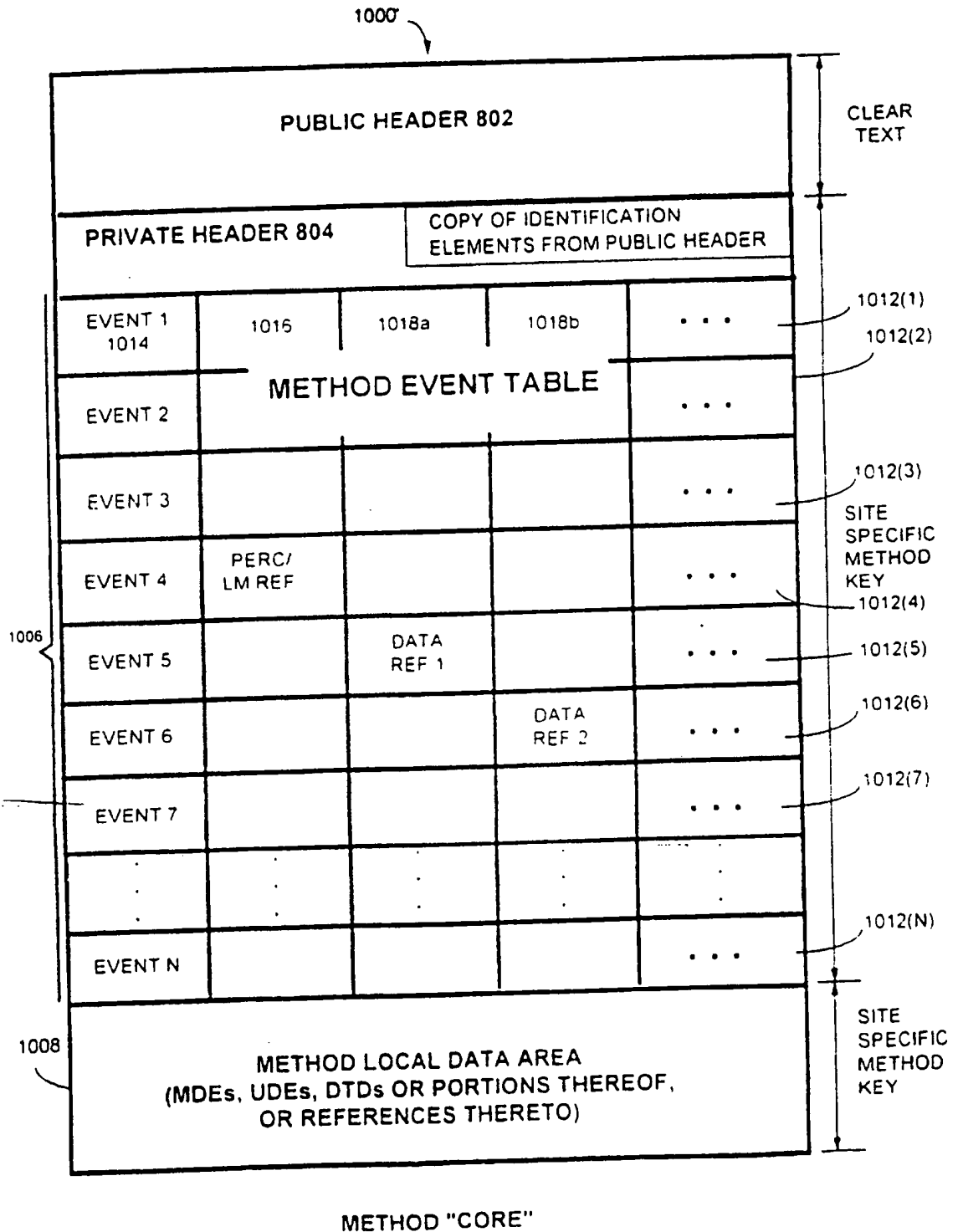


ADMINISTRATIVE OBJECT

FIG. 21

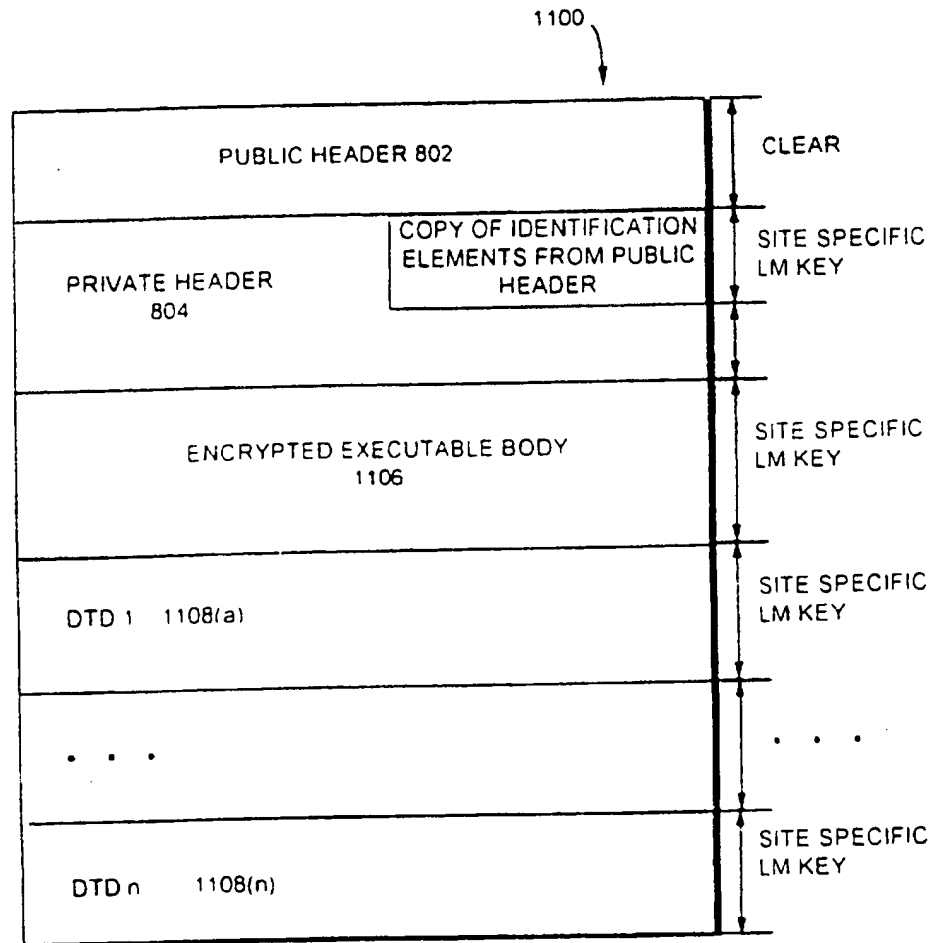
36/163

FIG. 22



37/163

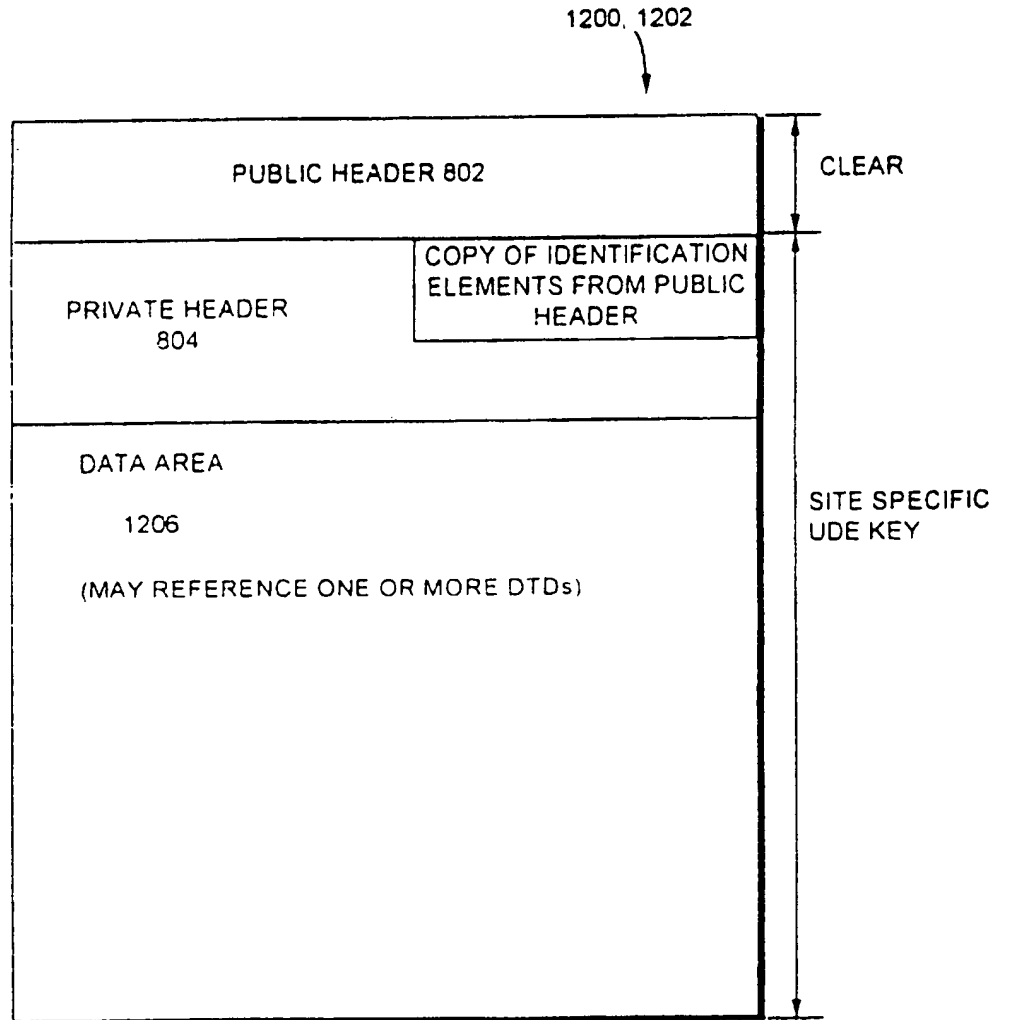
FIG. 23



LOAD MODULE

38/163

FIG. 24



UDE (MDE)



39/163

FIG. 25A

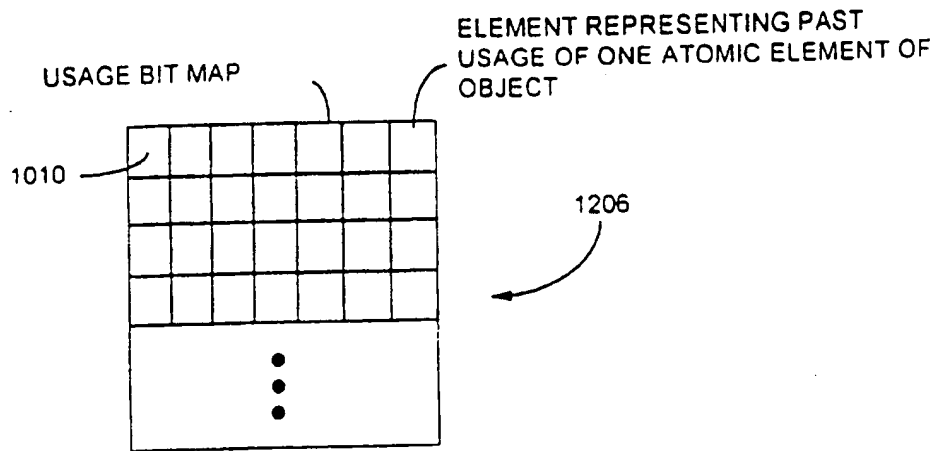
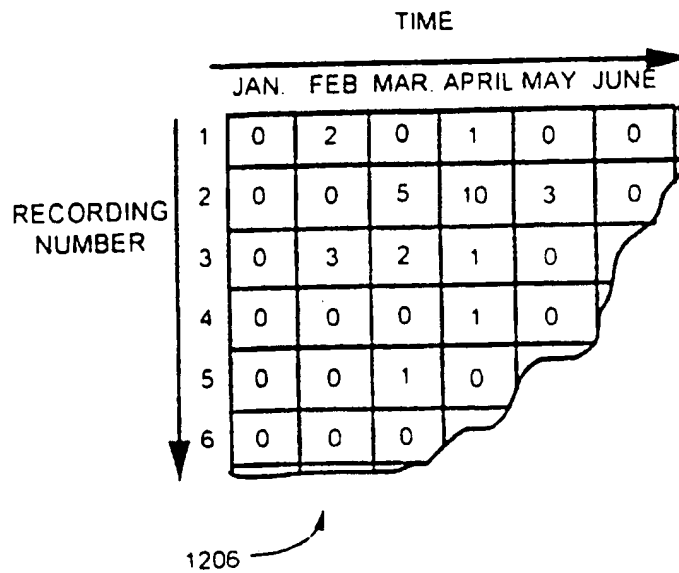


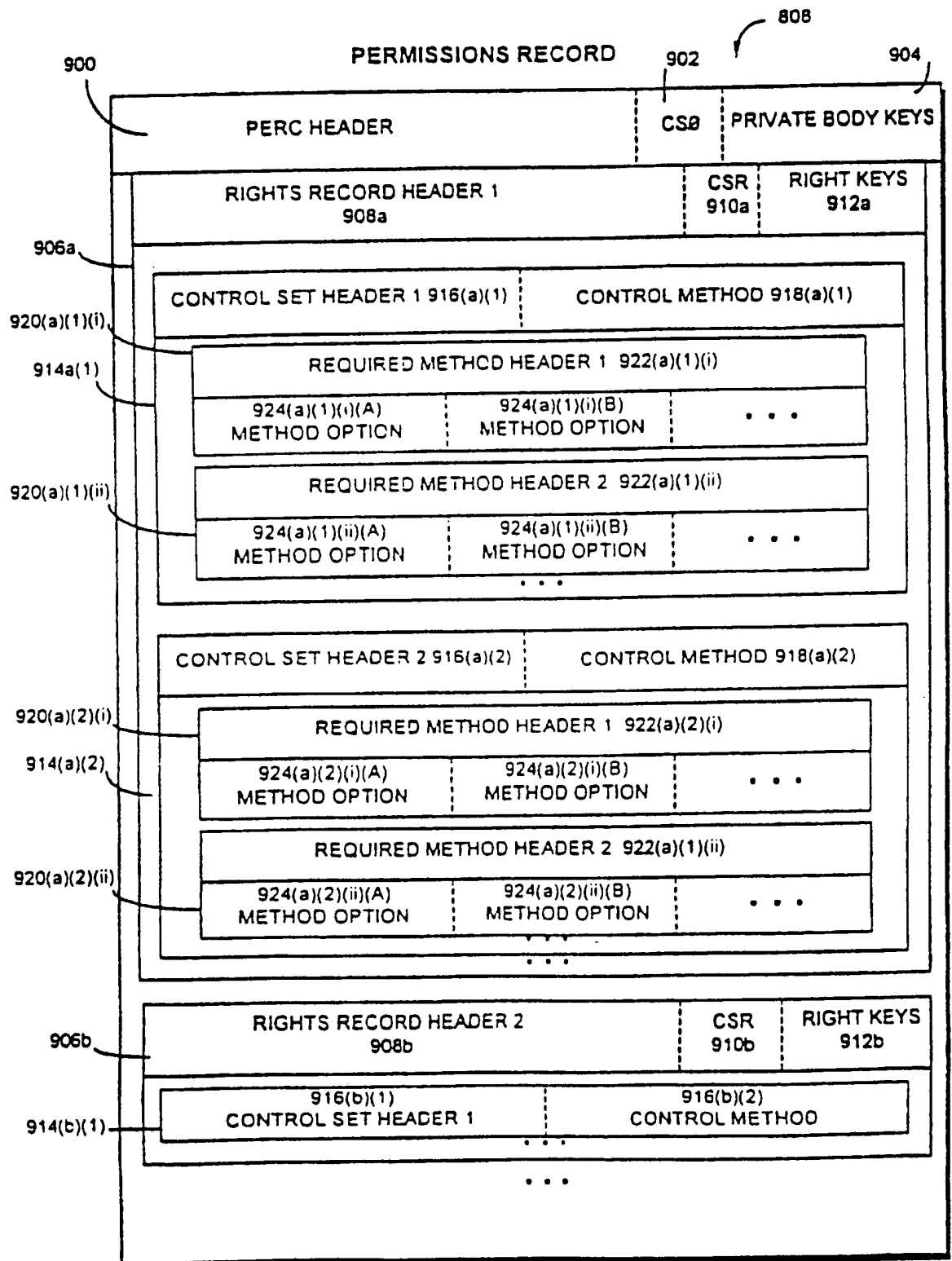
FIG. 25B





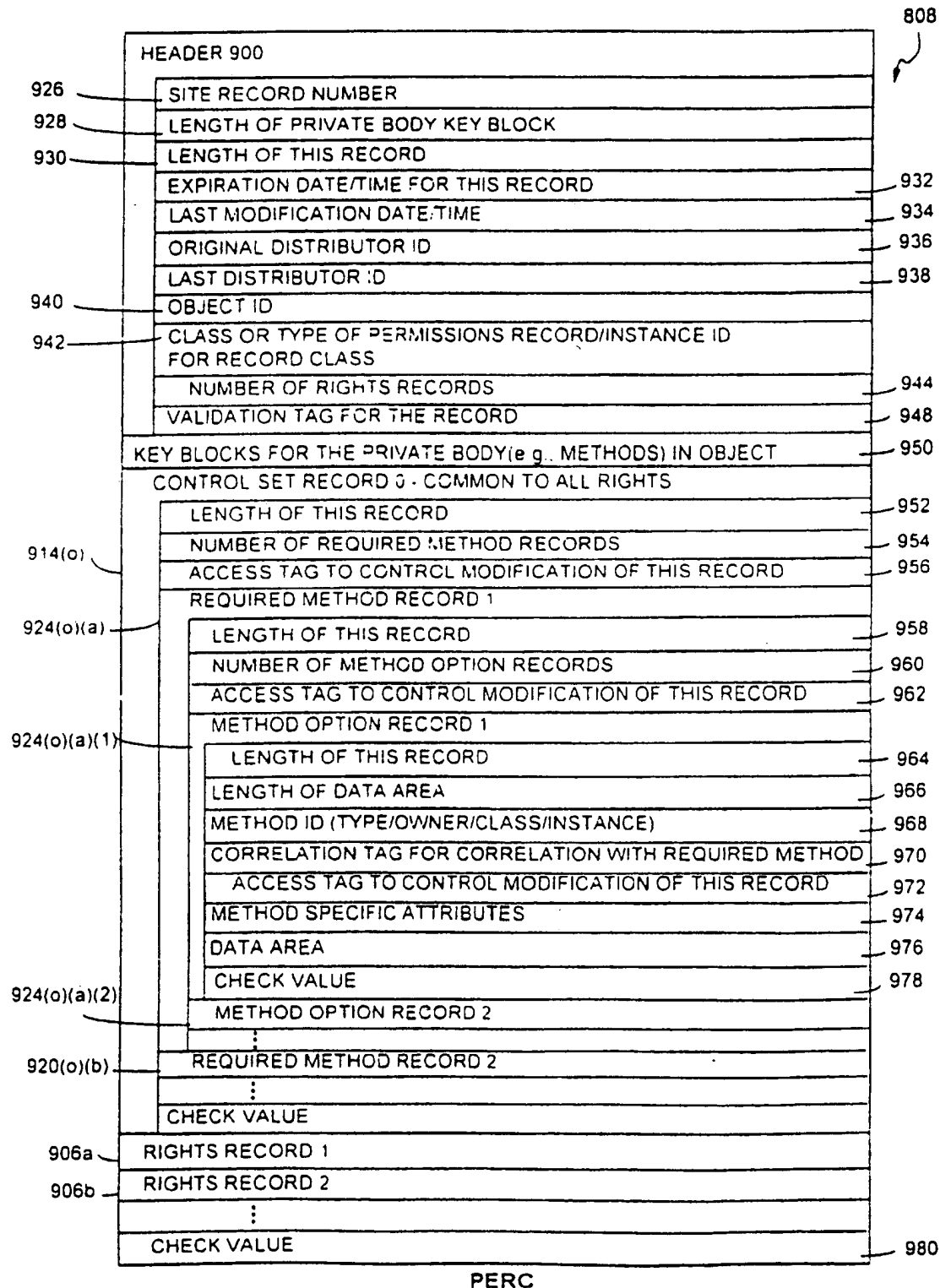
41/163

FIG. 26



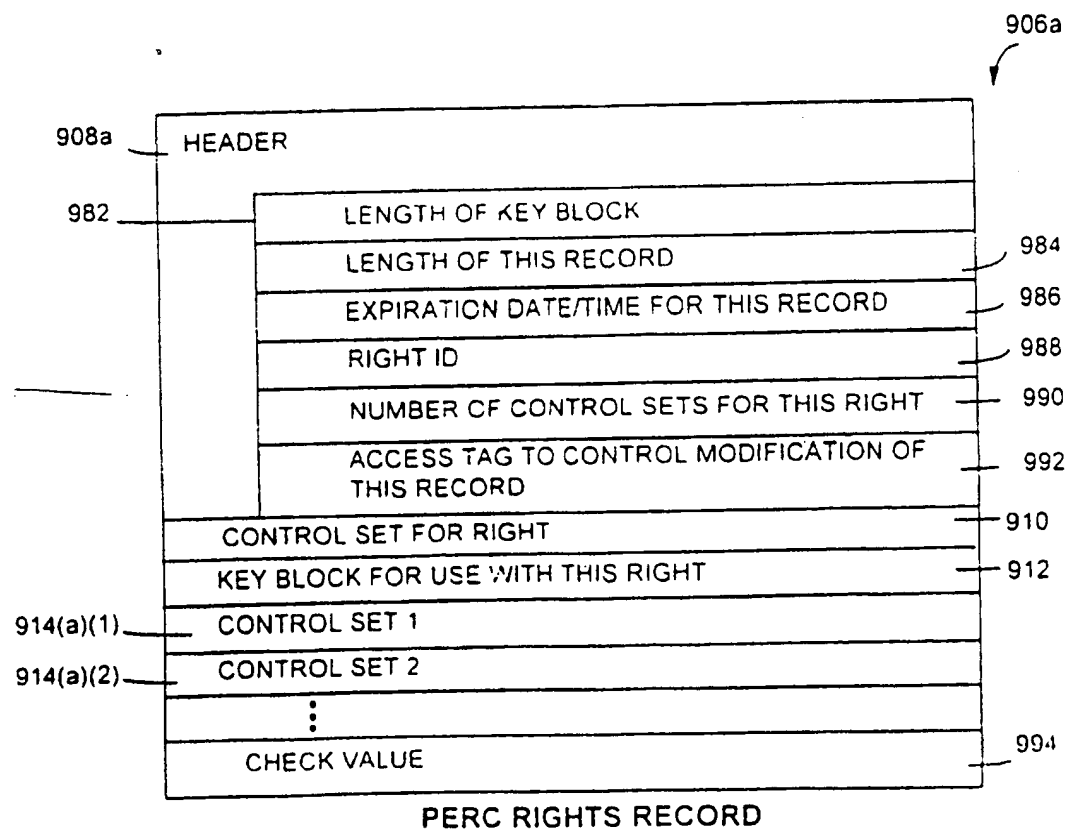
42/163

FIG. 26A



43/163

FIG. 26B



44/163

**FIG. 27**  
SHIPPING TABLE

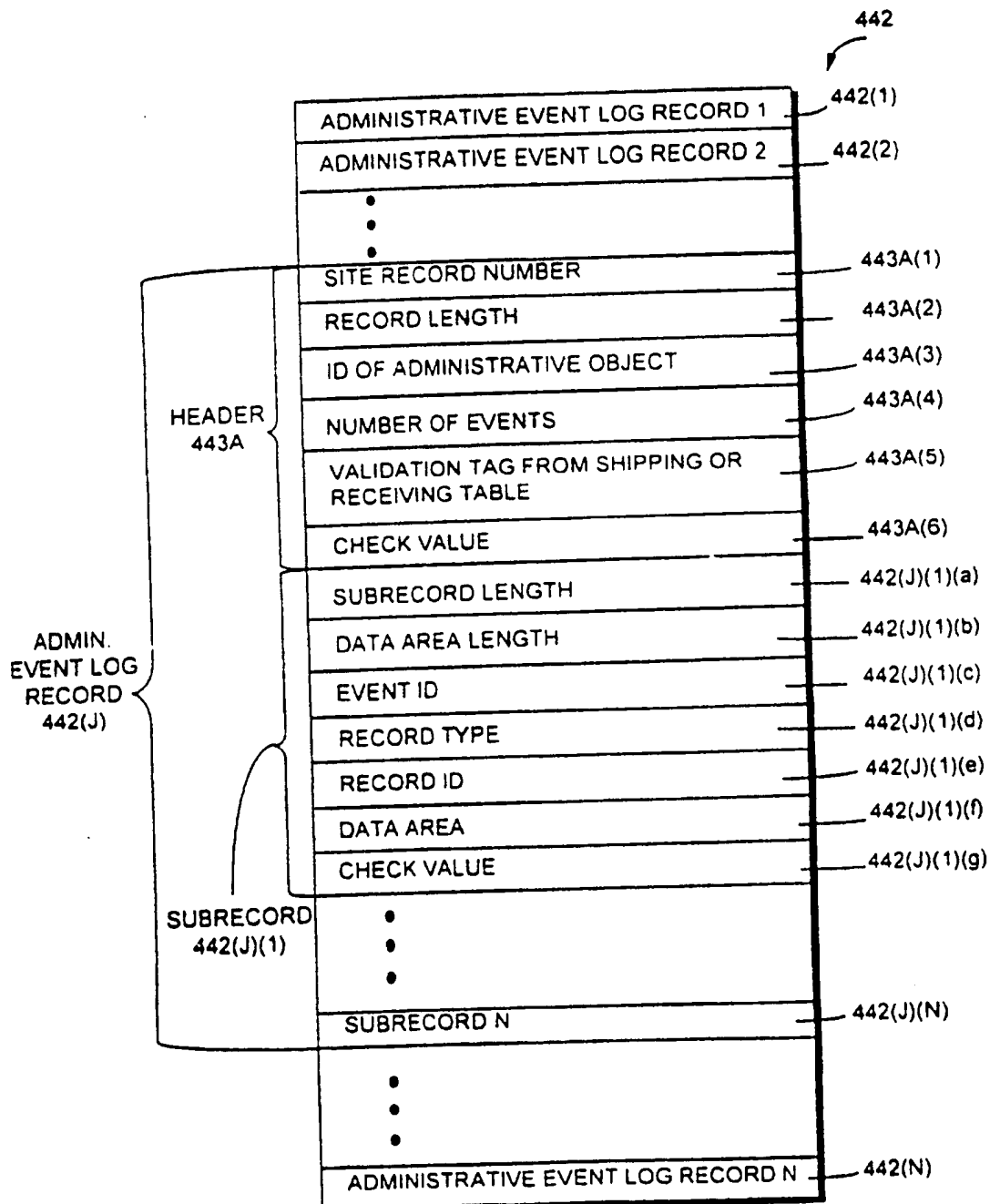
		444A(1)	
HEADER 444A	SITE RECORD NUMBER		444
	USER (GROUP) ID		444A(2)
	REF. TO "FIRST" COMPLETED OUTGOING SHIPPING RECORD		444A(3)
	REF. TO "LAST" COMPLETED OUTGOING SHIPPING RECORD		444A(4)
	REF. TO "FIRST" SCHEDULED OUTGOING SHIPPING RECORD		444A(5)
	REF. TO "LAST" SCHEDULED OUTGOING SHIPPING RECORD		444A(6)
	VALIDATION TAG FROM NAME SERVICES RECORD		444A(7)
	VALIDATION TAG FOR "FIRST" OUTGOING SHIPPING RECORD(S)		444A(8)
	CHECK VALUE		444A(9)
SHIPPING RECORD 445(1)	SITE RECORD NUMBER		445(1)(A)
	FIRST DATE/TIME FOR SCHEDULED SHIPMENT		445(1)(B)
	LAST DATE/TIME FOR SCHEDULED SHIPMENT		445(1)(C)
	ACTUAL DATE/TIME OF COMPLETED SHIPMENT		445(1)(D)
	OBJECT ID OF ADMINISTRATIVE OBJECT (TO BE) SHIPPED		445(1)(E)
	REF. TO ENTRY IN ADMINISTRATIVE EVENT LOG		445(1)(F)
	REF. TO NAME SERVICES RECORD NAMING RECIPIENT		445(1)(G)
	PURPOSE OF SHIPMENT		445(1)(H)
	STATUS OF SHIPMENT		445(1)(I)
	REF TO "PREVIOUS" OUTGOING SHIPPING RECORD		445(1)(J)
	REF. TO "NEXT" OUTGOING SHIPPING RECORD		445(1)(K)
	VALIDATION TAG FROM HEADER		445(1)(L)
	VALIDATION TAG TO ADMINISTRATIVE EVENT LOG		445(1)(M)
	VALIDATION TAG TO NAME SERVICES RECORD		445(1)(N)
	VALIDATION TAG FROM PREVIOUS RECORD		445(1)(O)
	VALIDATION TAG TO NEXT RECORD		445(1)(P)
	CHECK VALUE		445(1)(Q)
	⋮		
	SHIPPING RECORD N		445(1)(R)

45/163

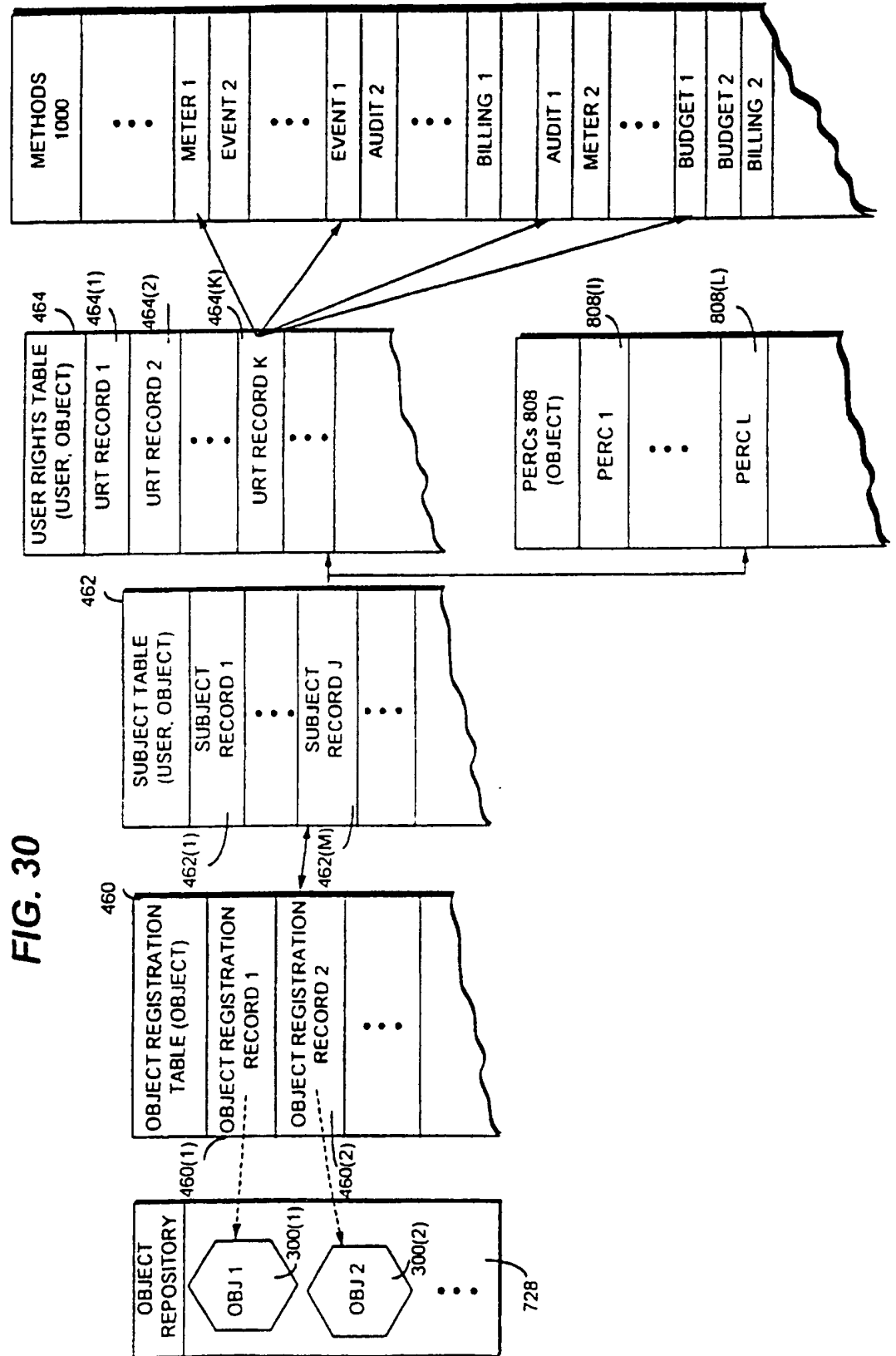
**FIG. 28**  
RECEIVING TABLE

		446A(1)	
HEADER 446A	SITE RECORD NUMBER		446
	USER (GROUP) ID		446A(2)
	REF. TO "FIRST" COMPLETED INCOMING RECEIVING RECORD		446A(3)
	REF. TO "LAST" COMPLETED INCOMING RECEIVING RECORD		446A(4)
	REF. TO "FIRST" SCHEDULED INCOMING RECEIVING RECORD		446A(5)
	REF. TO "LAST" SCHEDULED INCOMING RECEIVING RECORD		446A(6)
	VALIDATION TAG FROM NAME SERVICES RECORD		446A(7)
	VALIDATION TAG FOR "FIRST" INCOMING RECEIVING RECORD(S)		446A(8)
	CHECK VALUE		446A(9)
RECEIVING RECORD 447(1)	SITE RECORD NUMBER		447(1)(A)
	FIRST DATE/TIME FOR SCHEDULED RECEPTION		447(1)(B)
	LAST DATE/TIME FOR SCHEDULED RECEPTION		447(1)(C)
	ACTUAL DATE/TIME OF COMPLETED RECEPTION		447(1)(D)
	OBJECT ID OF ADMINISTRATIVE OBJECT (TO BE) RECEIVED		447(1)(E)
	REF. TO ENTRY IN ADMINISTRATIVE EVENT LOG		447(1)(F)
	REF. TO NAME SERVICES RECORD NAMING SENDER		447(1)(G)
	PURPOSE OF RECEPTION		447(1)(H)
	STATUS OF RECEPTION		447(1)(I)
	REF. TO "PREVIOUS" INCOMING RECEIVING RECORD		447(1)(J)
	REF. TO "NEXT" INCOMING RECEIVING RECORD		447(1)(K)
	VALIDATION TAGS		447(1)(L)
	CHECK VALUE		447(1)(M)
	⋮		
	RECEIVING RECORD N		447(2)

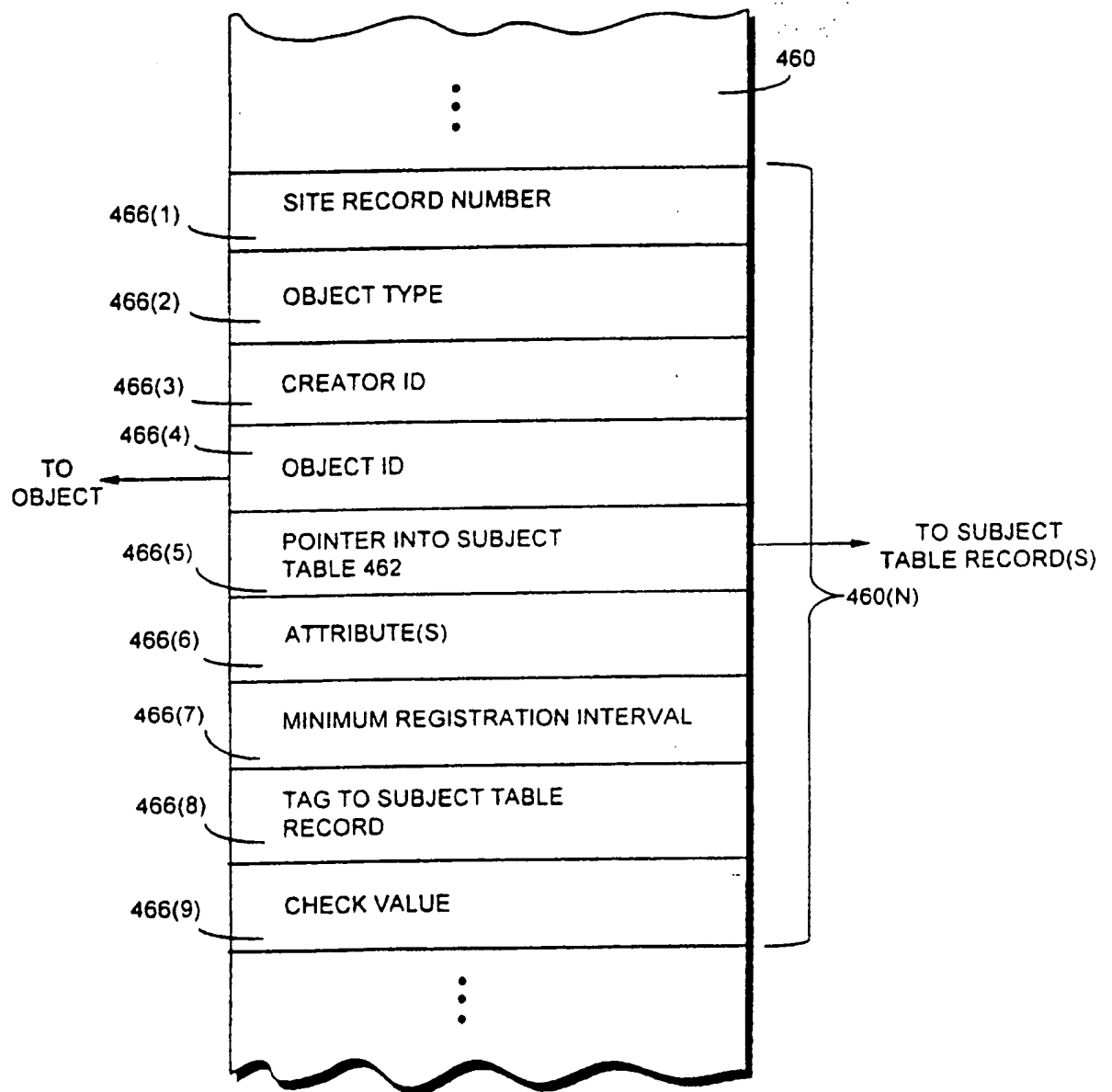
**FIG. 29**  
ADMINISTRATIVE EVENT LOG







48/163



**FIG. 31**  
OBJECT REGISTRATION TABLE  
SUBSTITUTE SHEET (RULE 26)

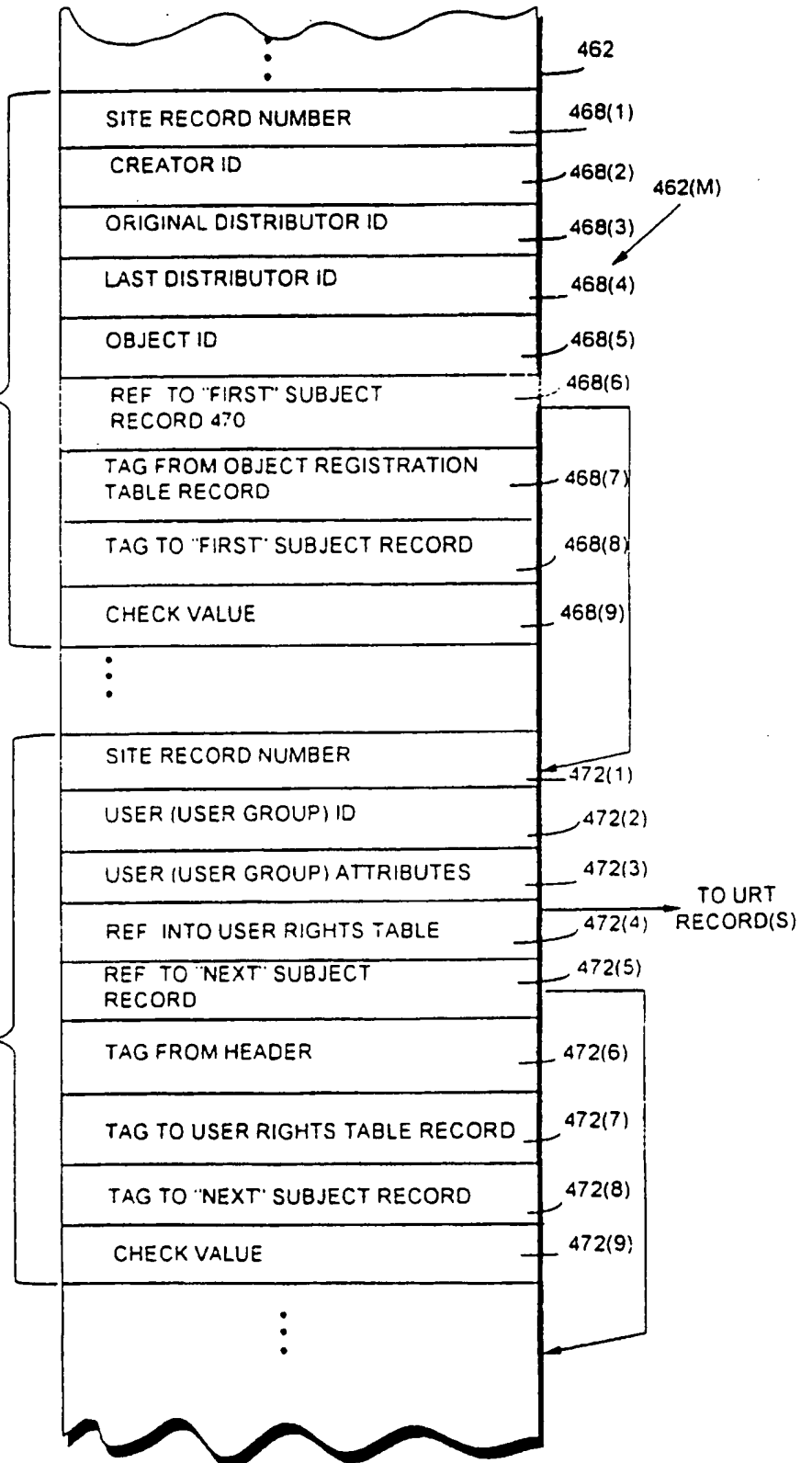
49/163

**FIG. 32**

SUBJECT  
TABLE

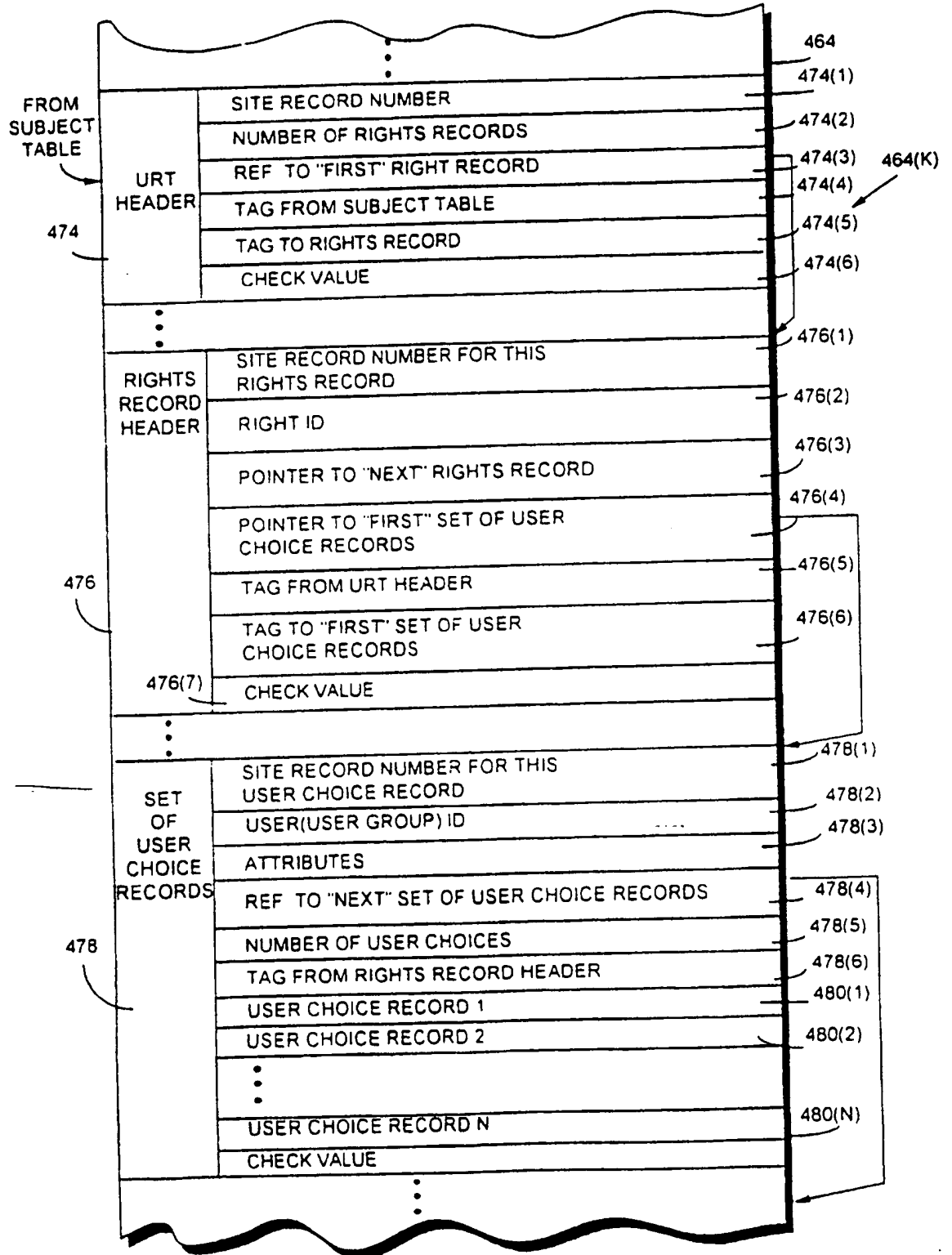
"HEADER"  
468

SUBJECT  
RECORD  
470(1)

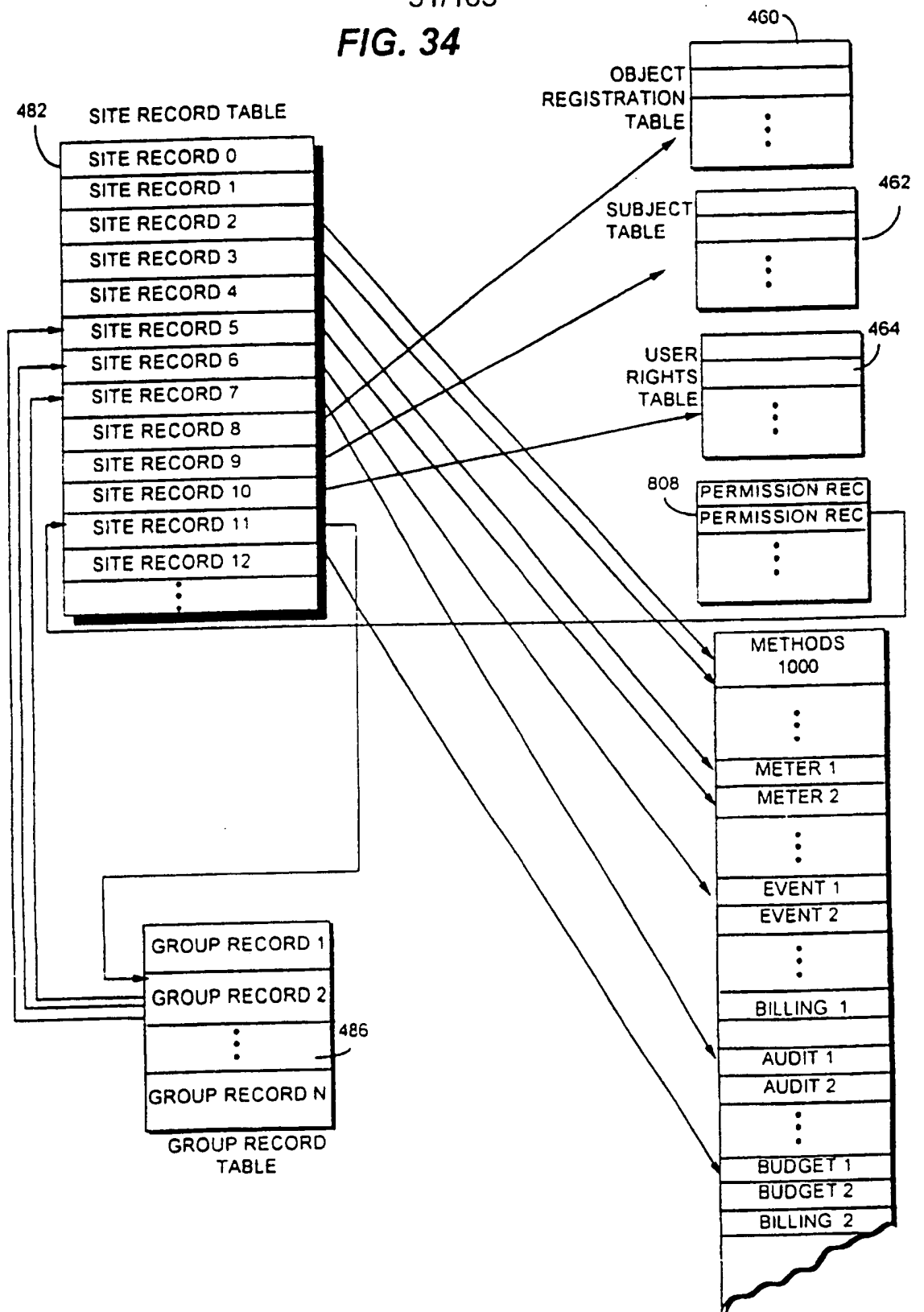


50/163

**FIG. 33** USER RIGHTS TABLE



51/163  
**FIG. 34**



52/163

**FIG. 34A****SITE RECORD**

The diagram shows a vertical stack of fields representing a 'SITE RECORD'. The top field is a header labeled '482(J)' and is part of a larger structure labeled '482'. Below the header are ten data fields, each with a label on the left and a reference number on the right. The fields are: 'TYPE OF RECORD' (484(1)), 'OWNER OR CREATOR OF RECORD' (484(2)), 'CLASS' (484(3)), 'INSTANCE' (484(4)), 'TYPE SPECIFIC DESCRIPTOR (e.g., OBJECT ID) ASSOCIATED WITH RECORD' (484(5)), 'TABLE IN WHICH THE RECORD IS LOCATED' (484(6)), 'POINTER - OFFSET, WITHIN THE TABLE, TO WHERE THE RECORD BEGINS' (484(7)), 'RECORD LENGTH' (484(8)), 'VALIDATION TAG FOR RECORD' (484(9)), and 'CHECK VALUE' (484(10)). The bottom of the structure is wavy, indicating it is a partial view.

482	
482(J)	
TYPE OF RECORD	484(1)
OWNER OR CREATOR OF RECORD	484(2)
CLASS	484(3)
INSTANCE	484(4)
TYPE SPECIFIC DESCRIPTOR (e.g., OBJECT ID) ASSOCIATED WITH RECORD	484(5)
TABLE IN WHICH THE RECORD IS LOCATED	484(6)
POINTER - OFFSET, WITHIN THE TABLE, TO WHERE THE RECORD BEGINS	484(7)
RECORD LENGTH	484(8)
VALIDATION TAG FOR RECORD	484(9)
CHECK VALUE	484(10)

53/163

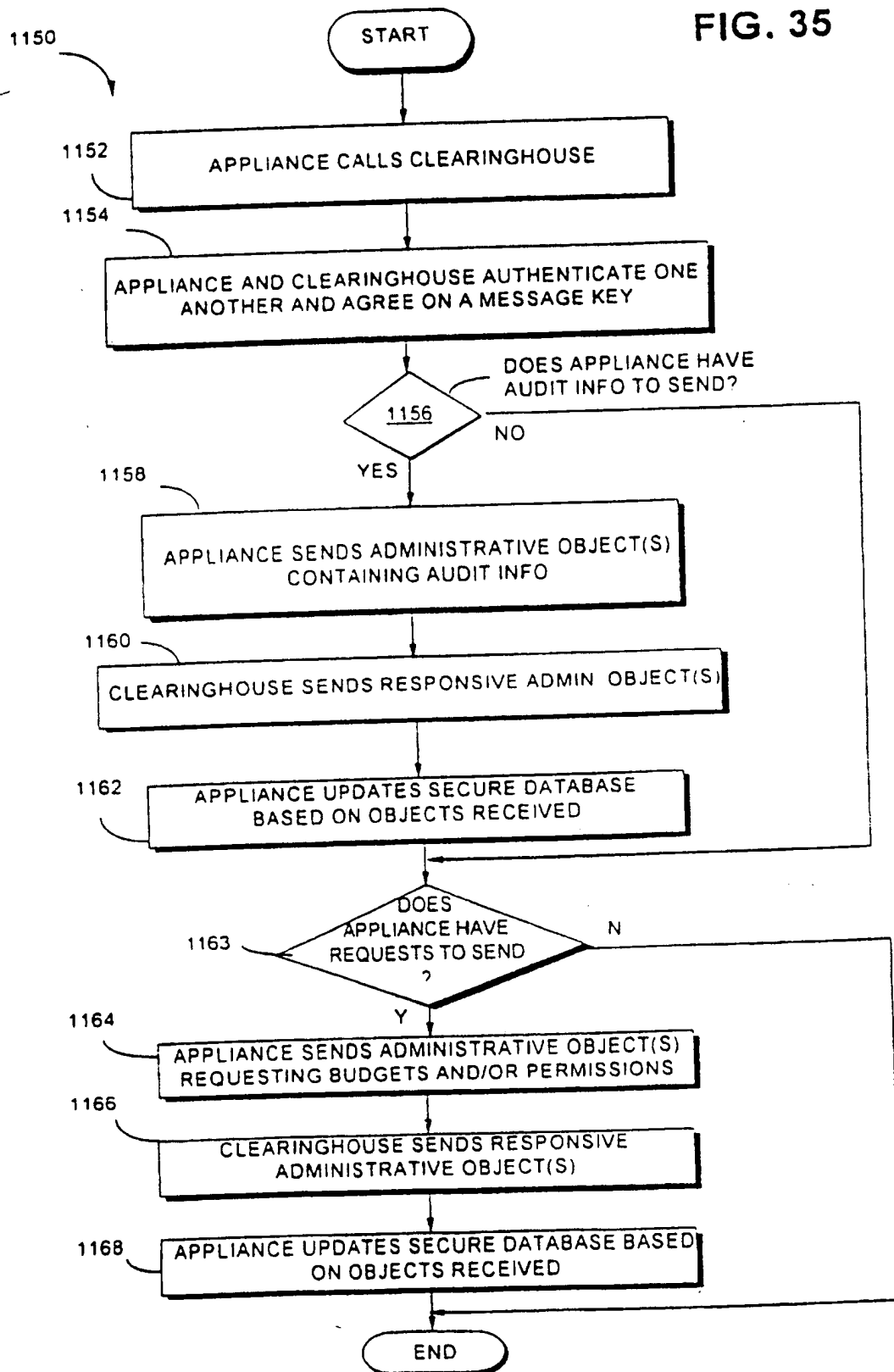
**FIG. 34B**

**GROUP RECORD**

		486
		486(J)
SITE RECORD NUMBER		488(1)
NUMBER OF REFERENCE SUBRECORDS		488(2)
VALIDATION TAG FOR GROUP OF RECORDS		488(3)
REFERENCE SUBRECORD 1		488(4)
REF (SITE RECORD NUMBER 1) FOR 1ST RECORD IN GROUP		490(A)
VALIDATION TAG FOR RECORD		490(B)
REFERENCE SUBRECORD 2		488(5)
REF (SITE RECORD NUMBER 2) FOR 1ST RECORD IN GROUP		490(C)
VALIDATION TAG FOR RECORD		490(D)
:		
CHECKSUM (CRC)		488(6)

54/163

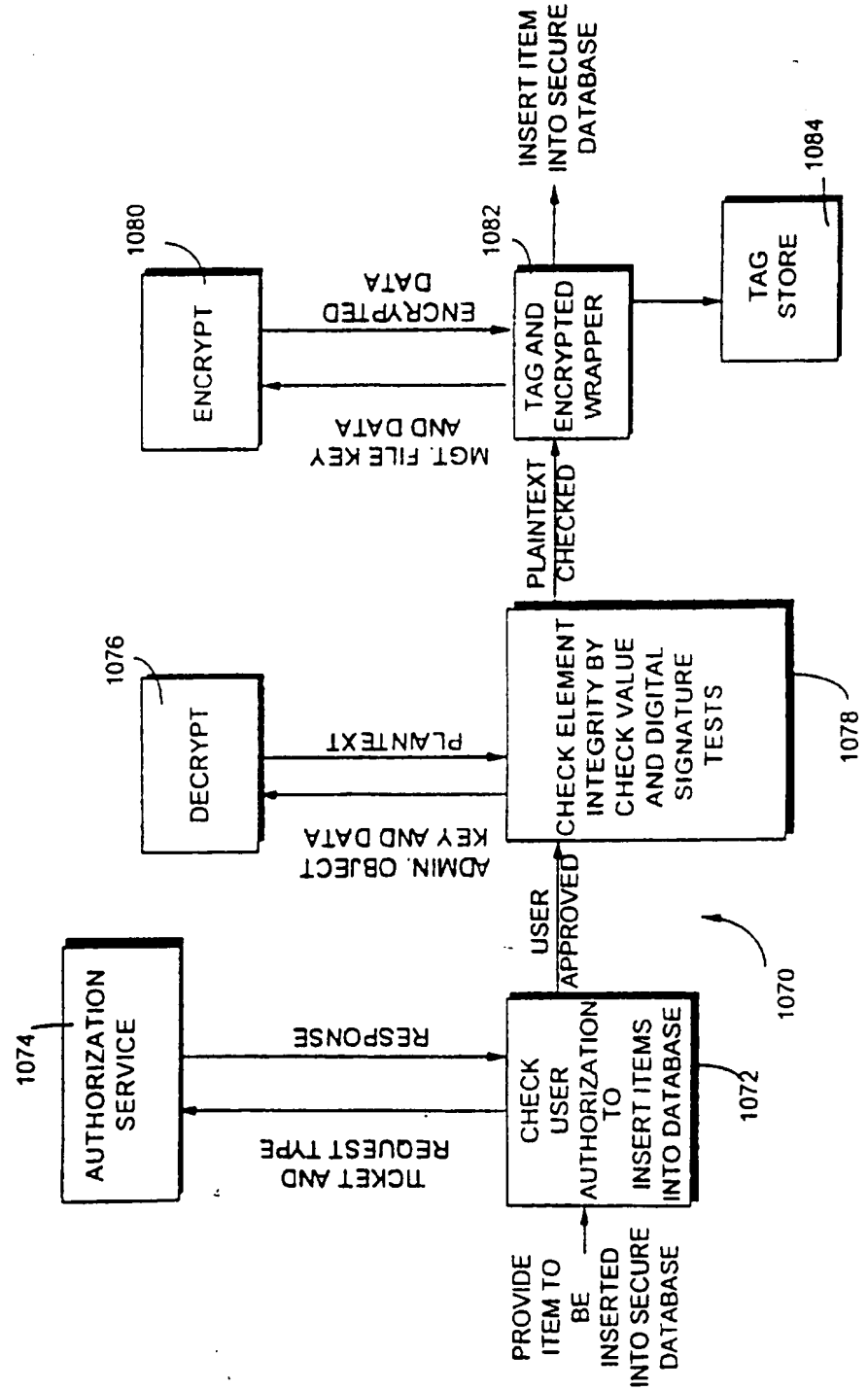
FIG. 35





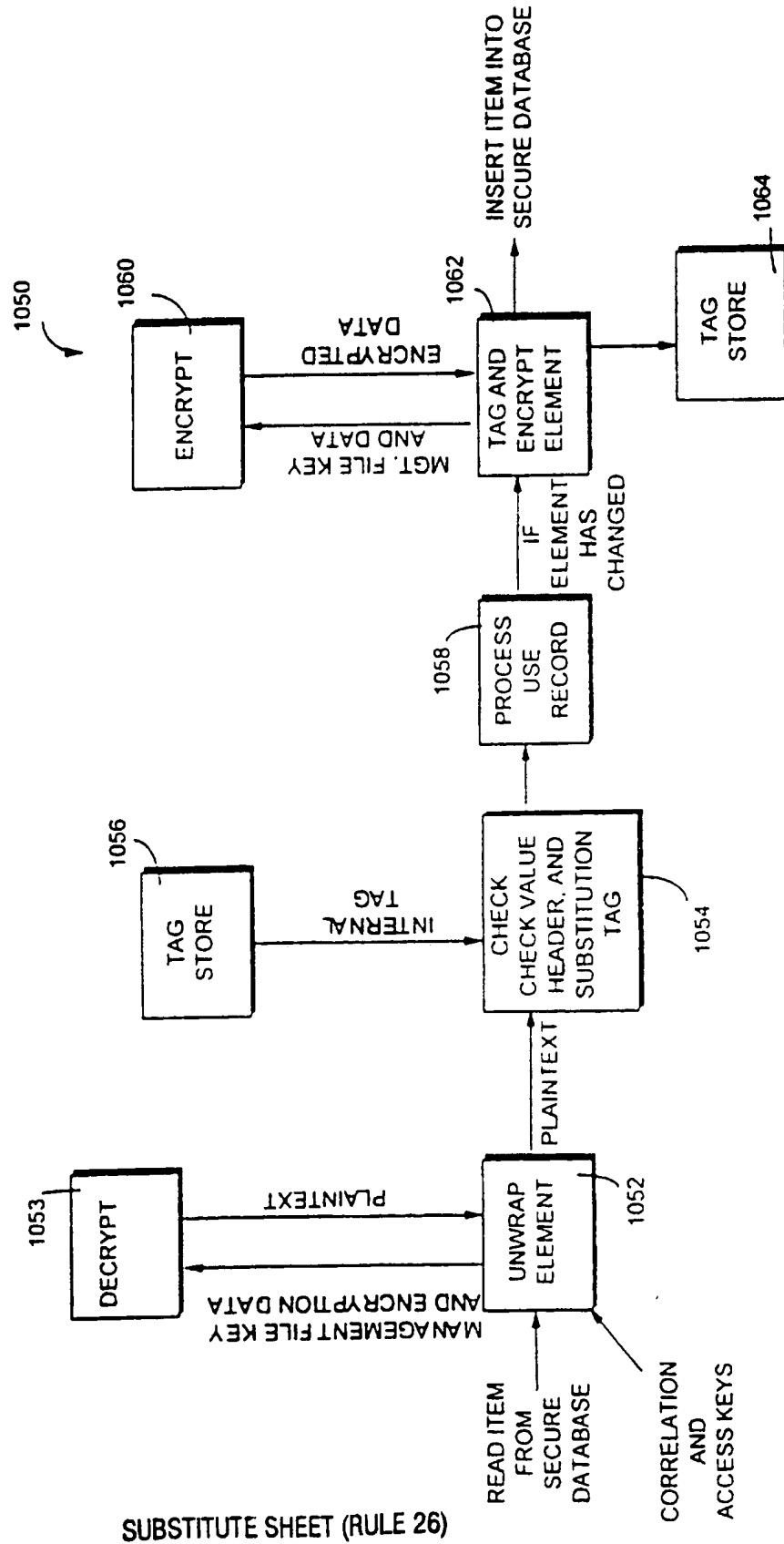
55/163

FIG. 36



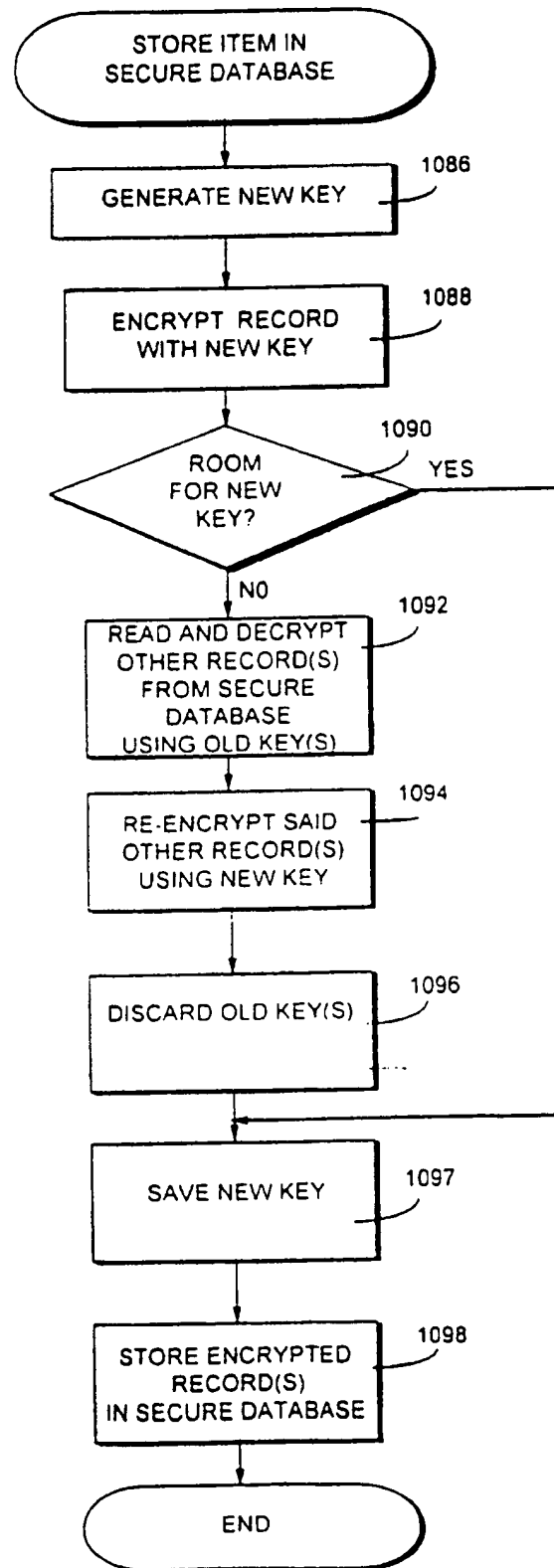
56/163

FIG. 37

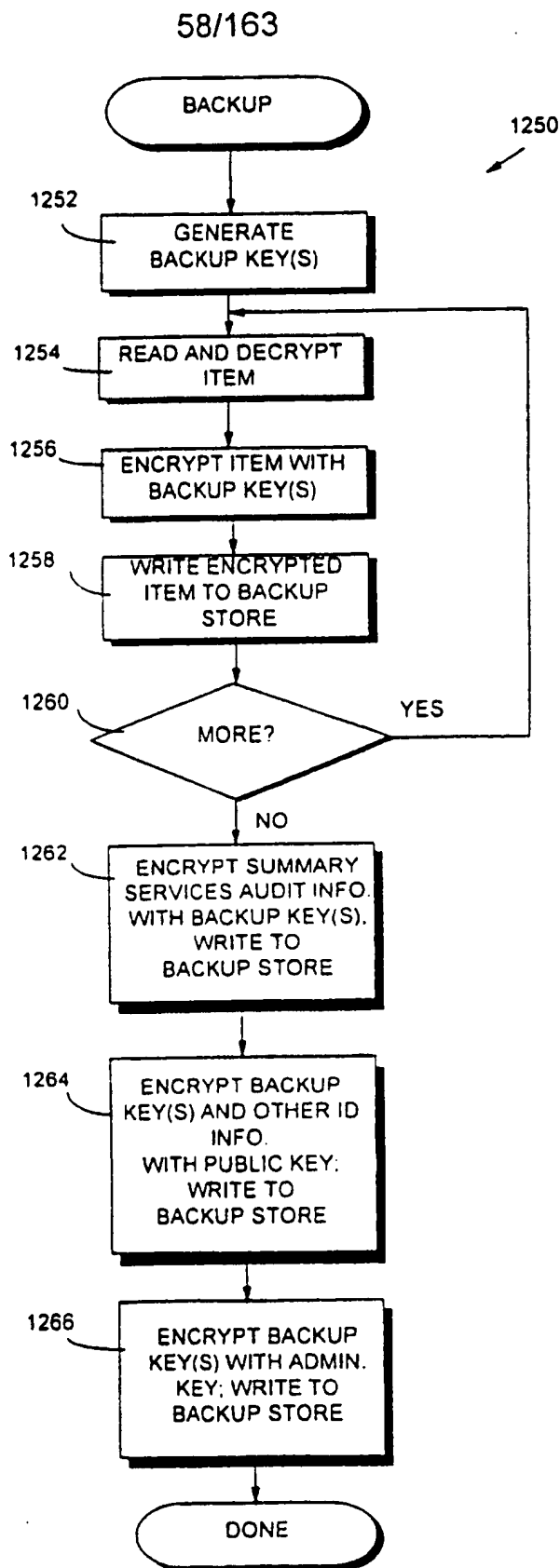


57/163

FIG. 38



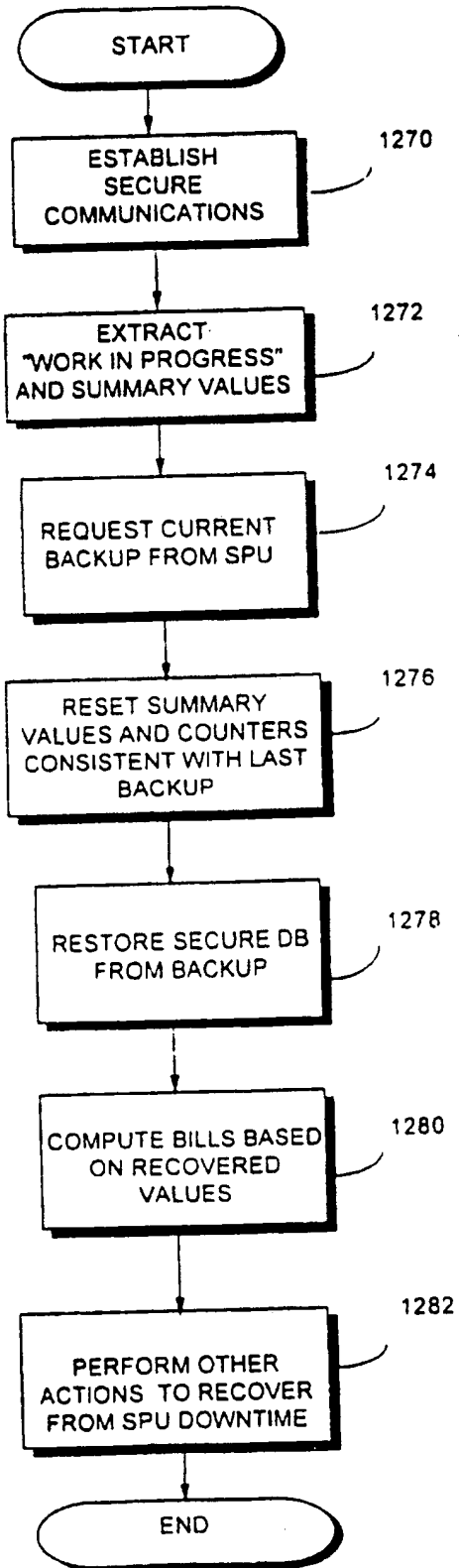
**FIG. 39**  
BACKUP



59/163

**FIG. 40**  
RECOVER SECURE DATABASE

1268



60/163

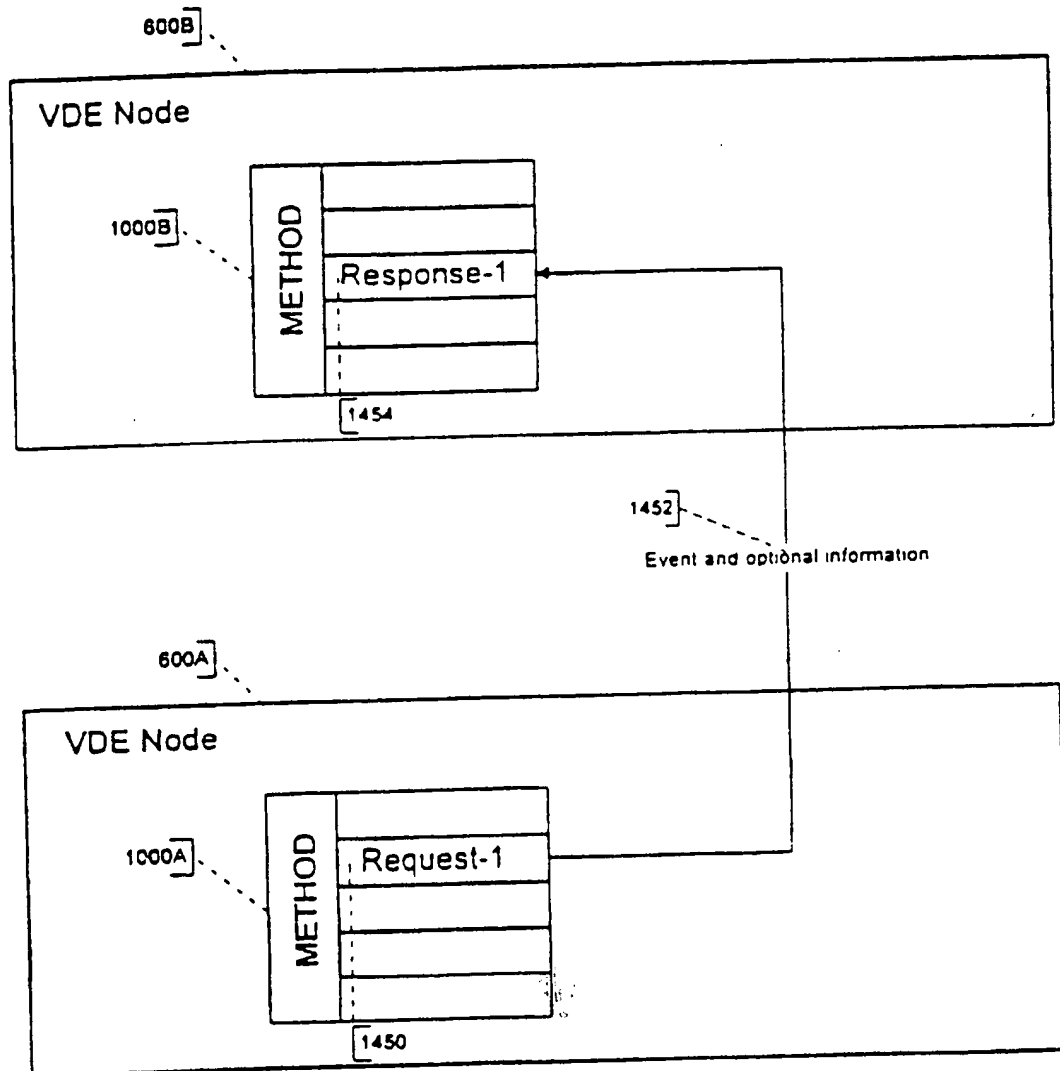


Figure 41a

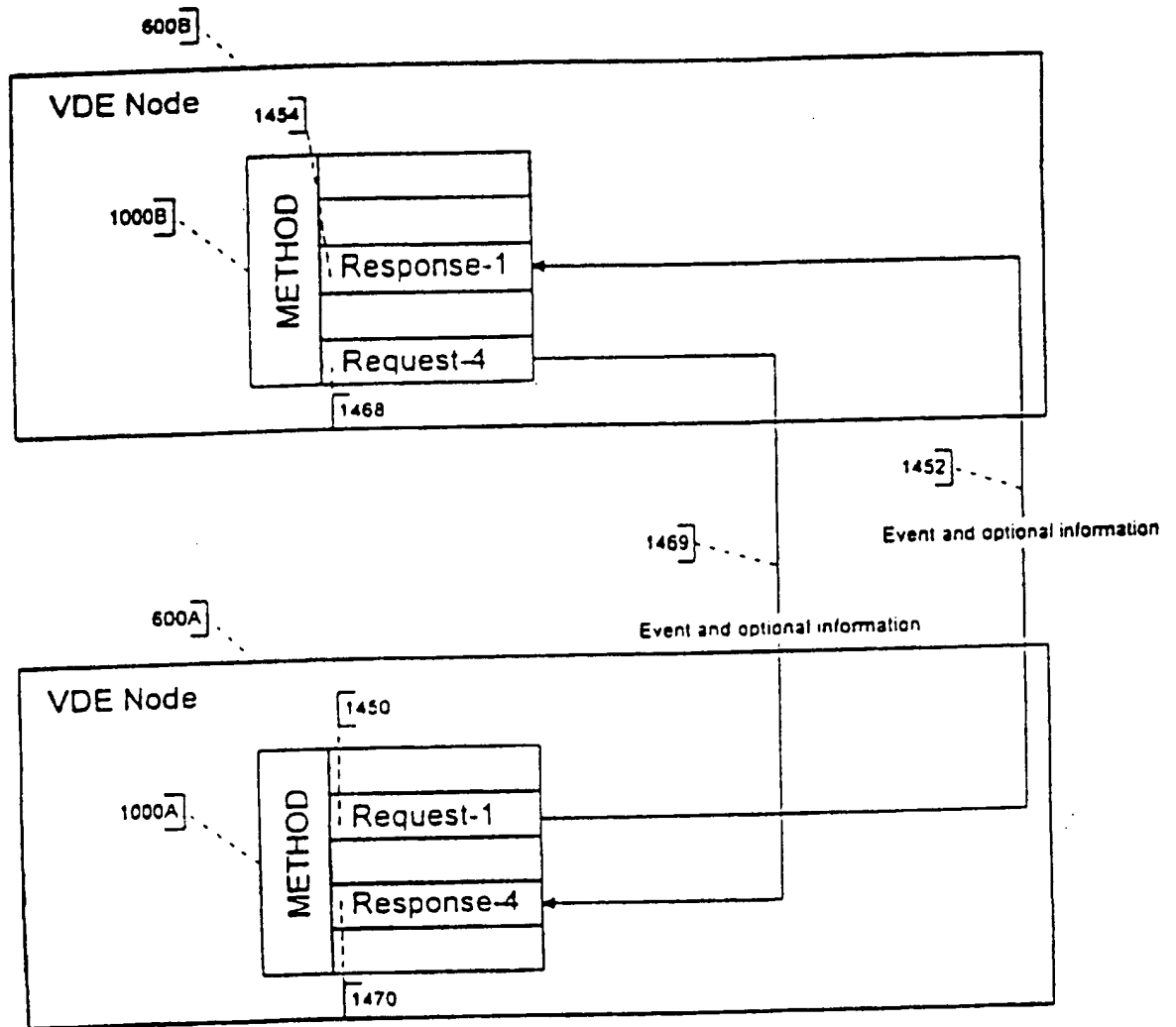


Figure 41b

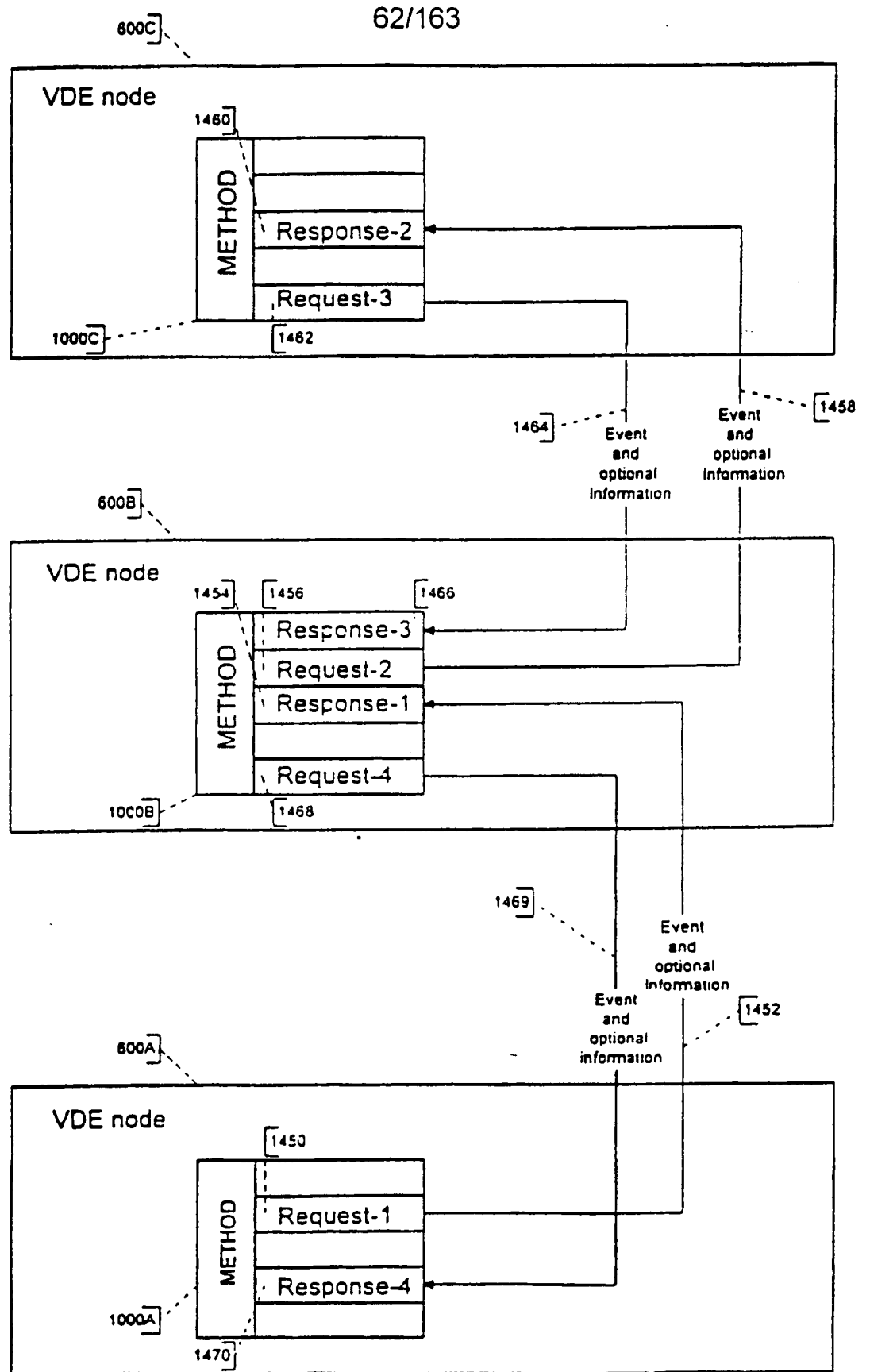


Figure 41c



63/163

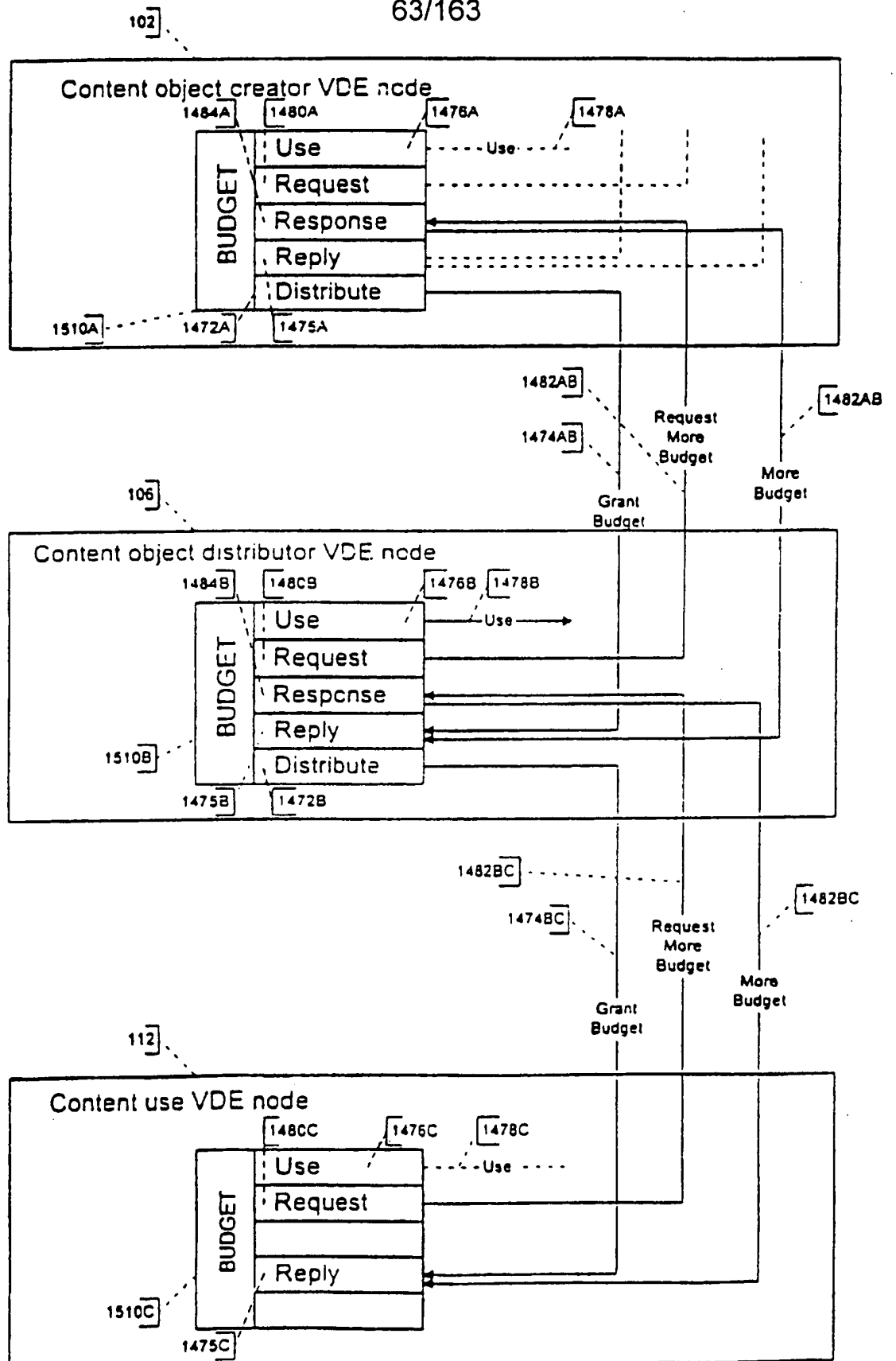


Figure 41d

64/163

# BUDGET Method Use Process Flow

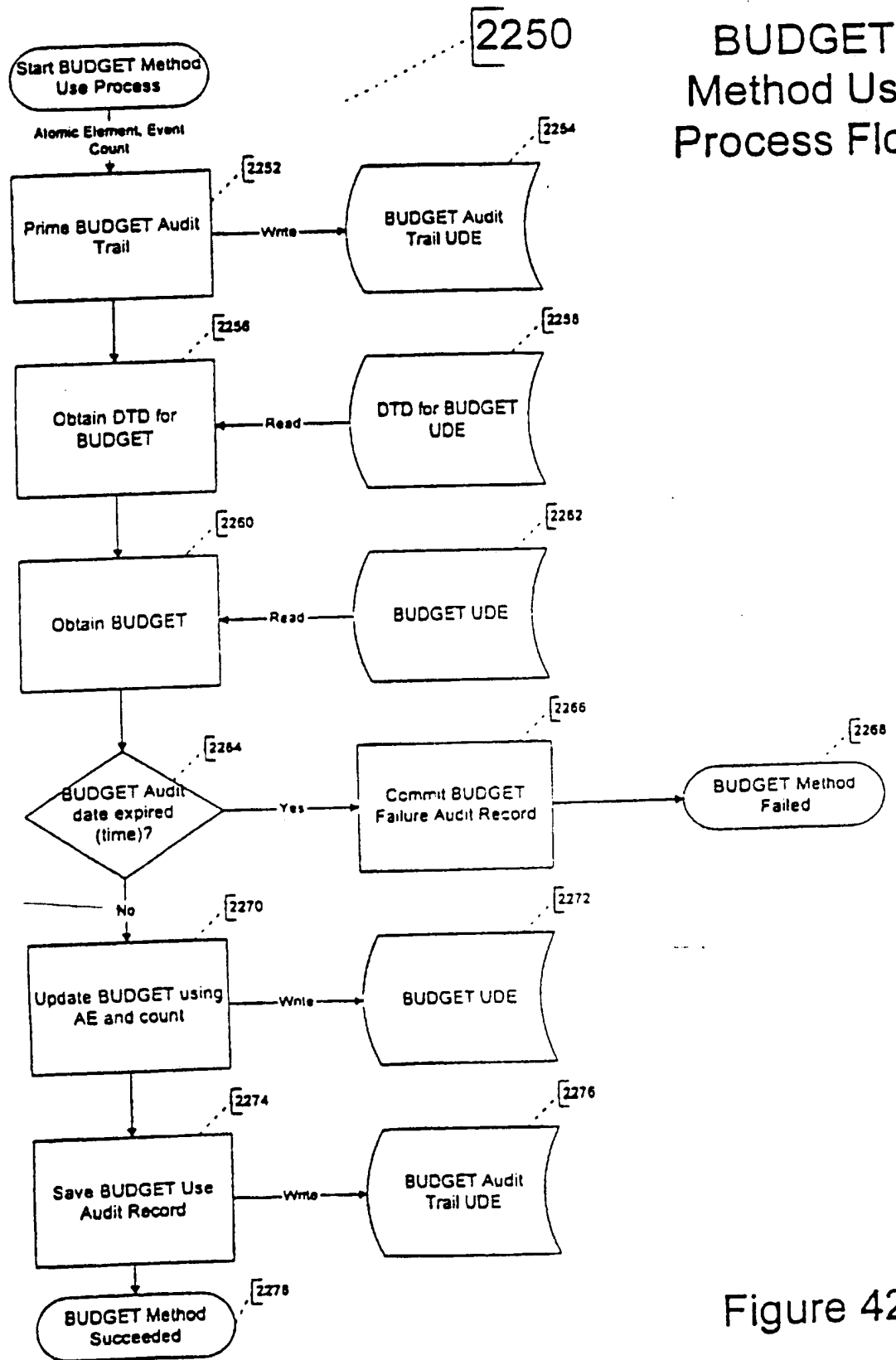


Figure 42a

65/163

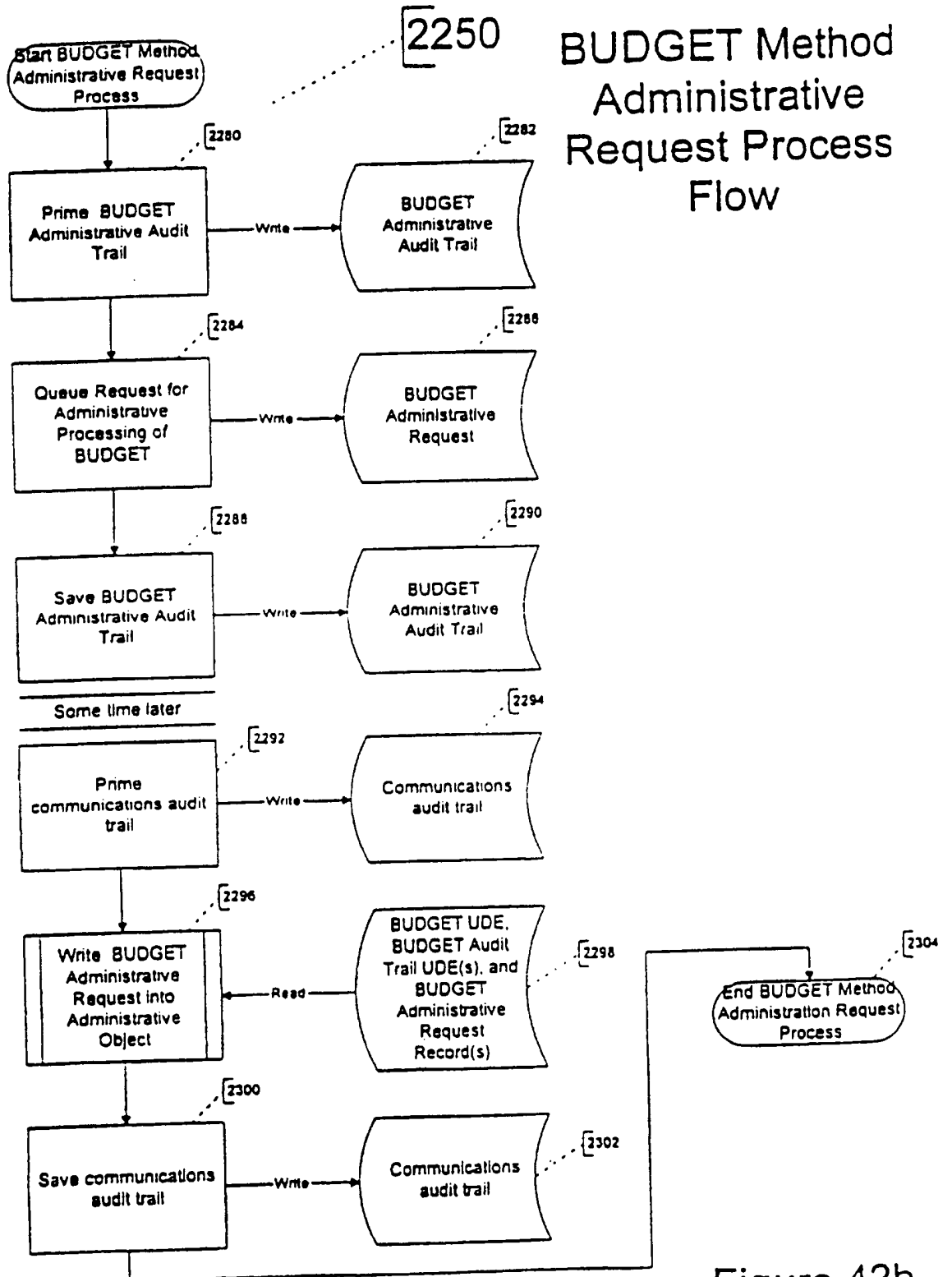


Figure 42b

66/163

# BUDGET Method Administrative Response Process Flow

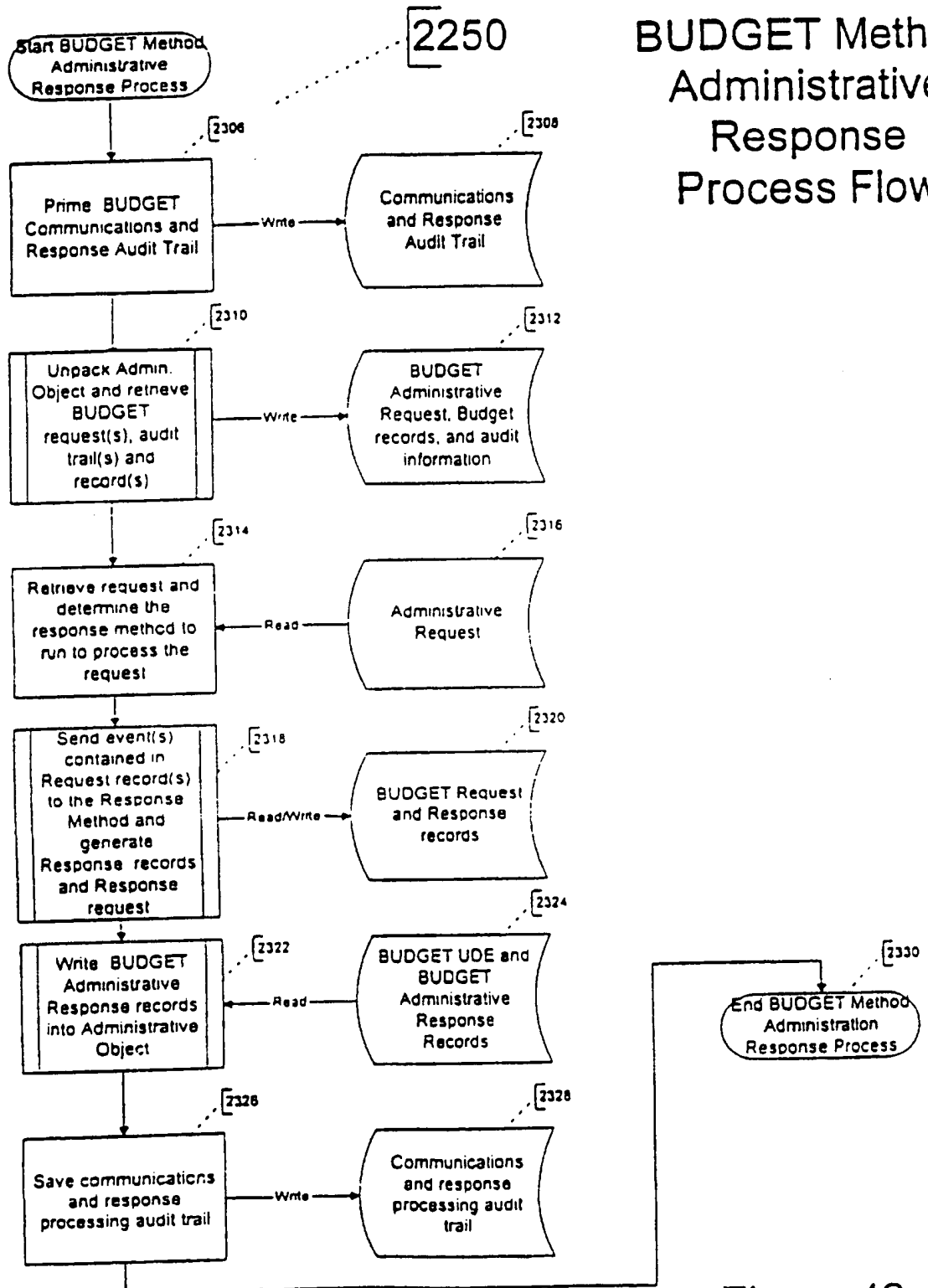


Figure 42c

67/163

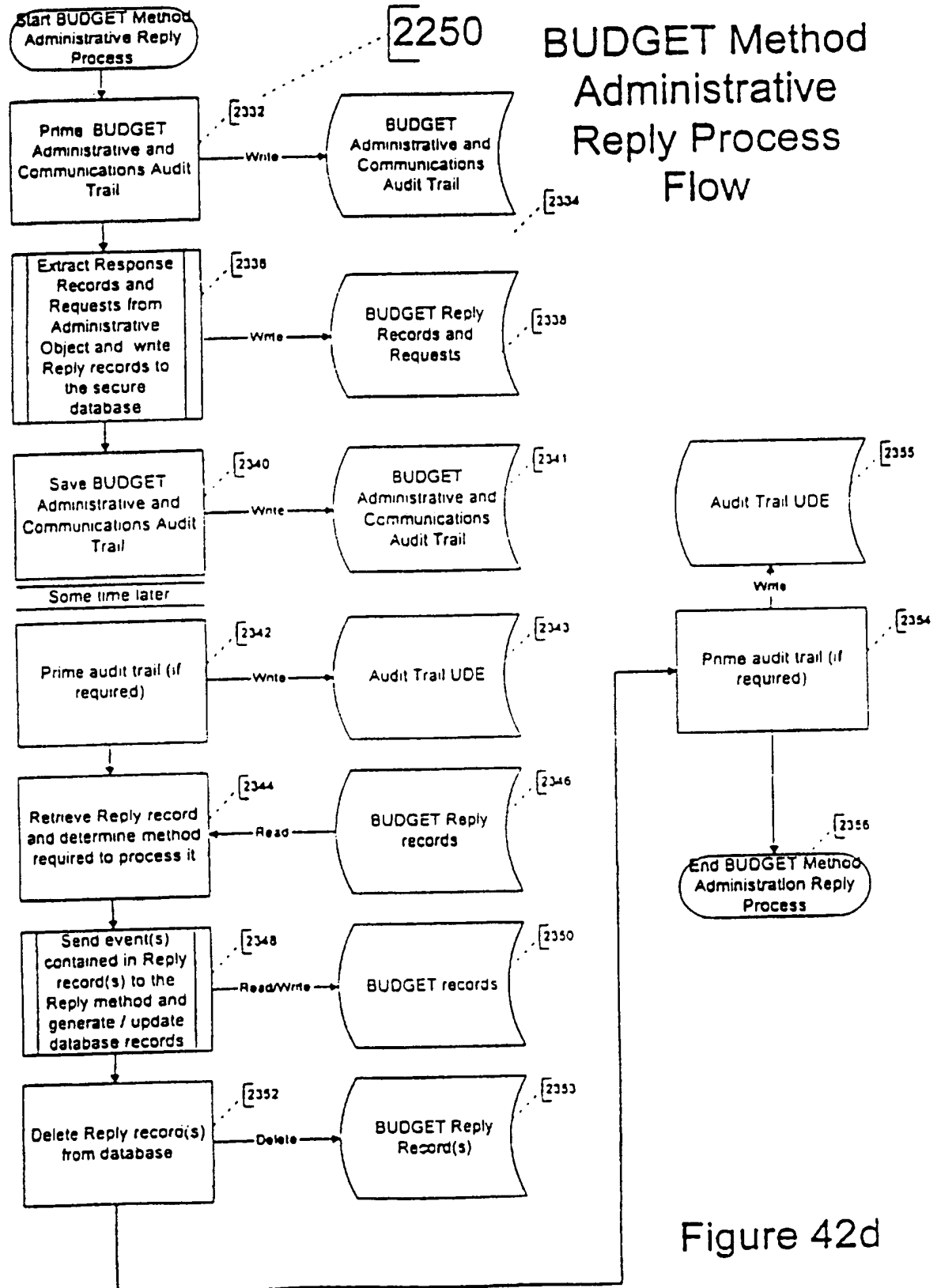


Figure 42d

68/163

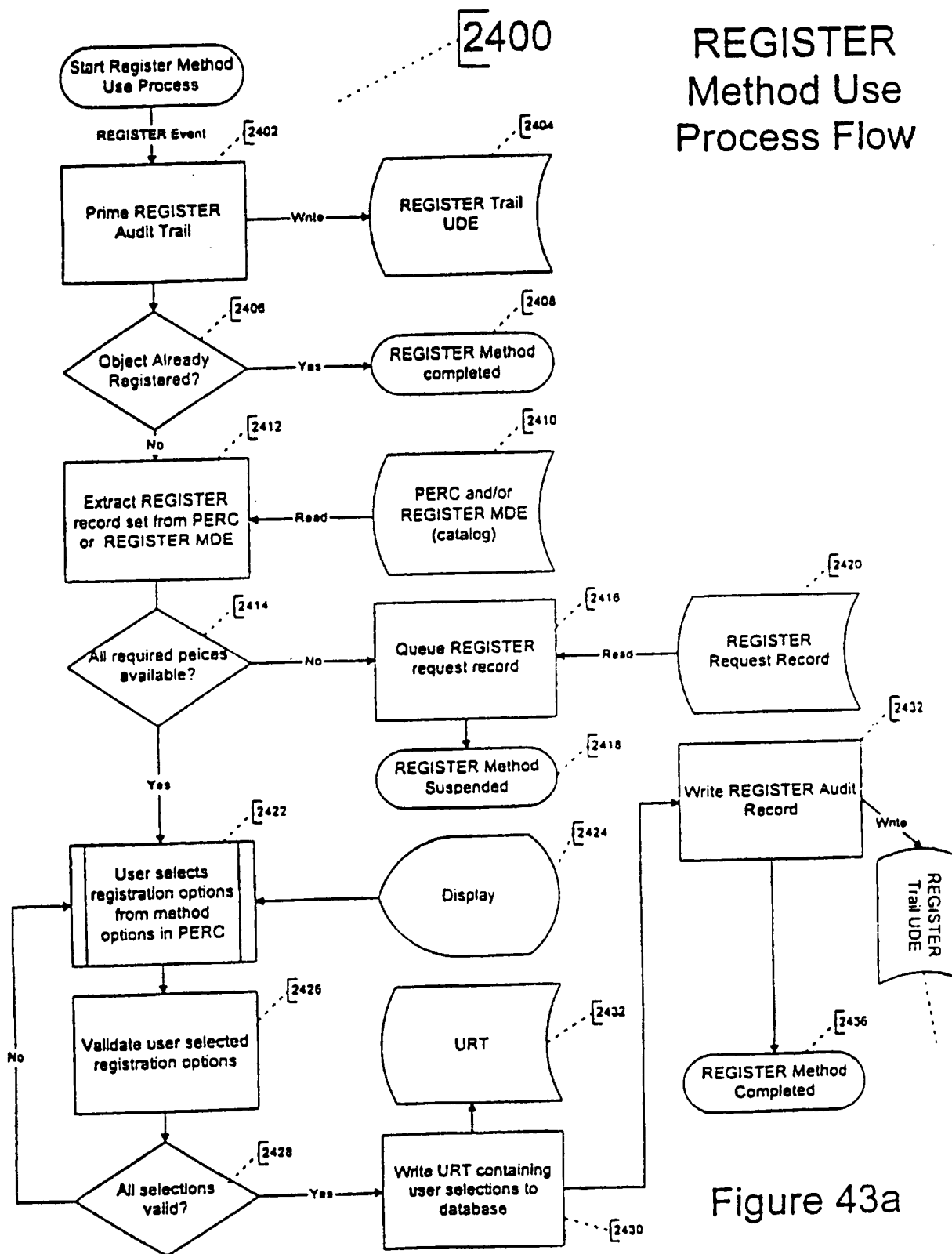


Figure 43a

# REGISTER Method Administrative Request Process Flow

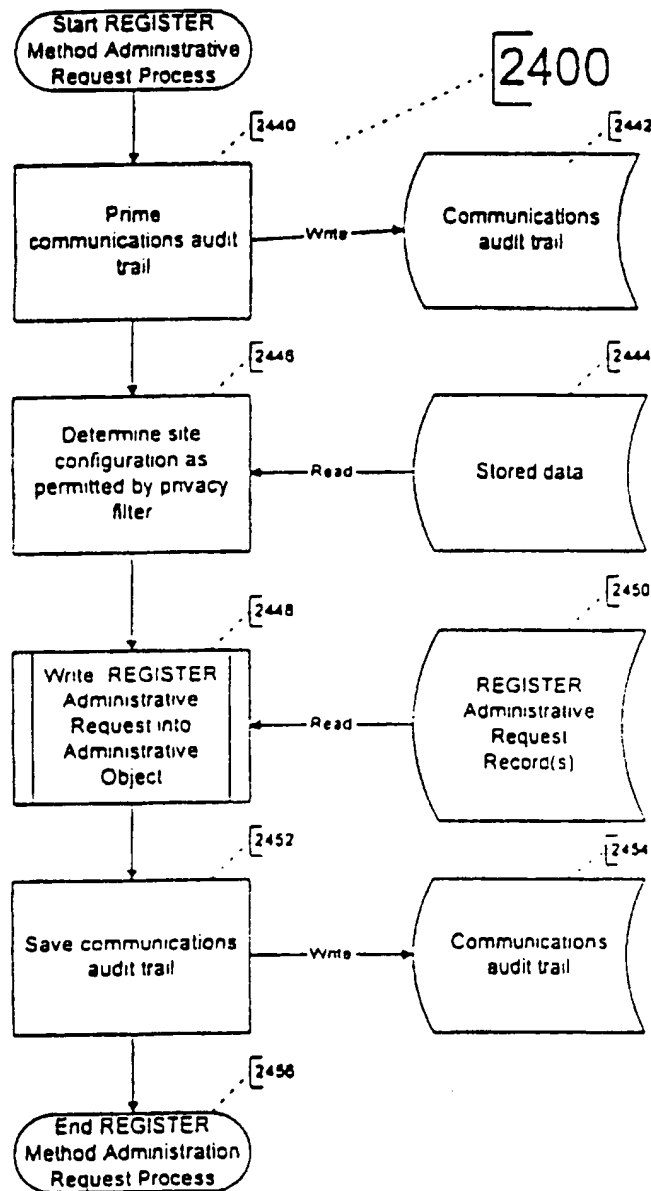
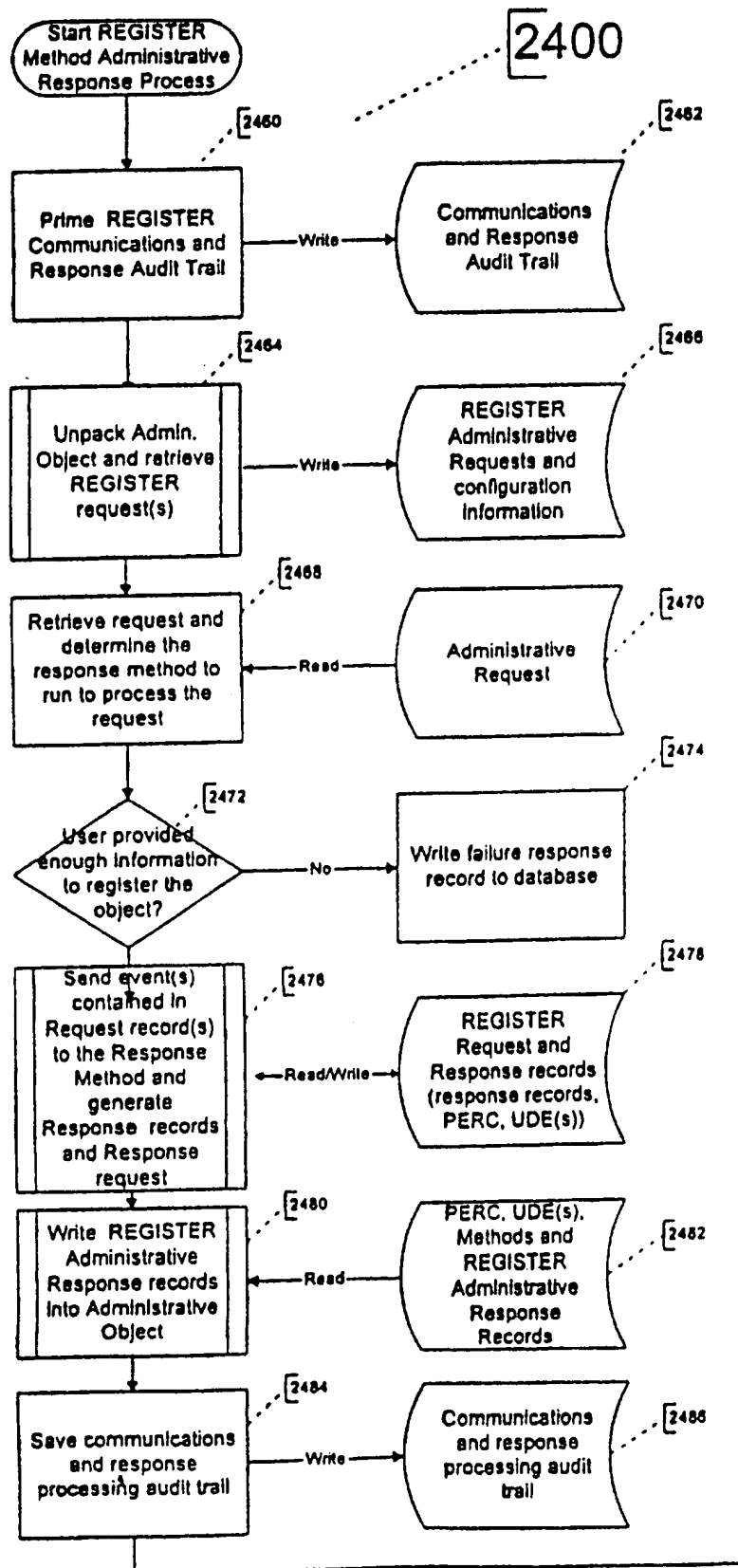


Figure 43b

70/163



# REGISTER Method Administrative Response Process Flow

Figure 43c



71/163

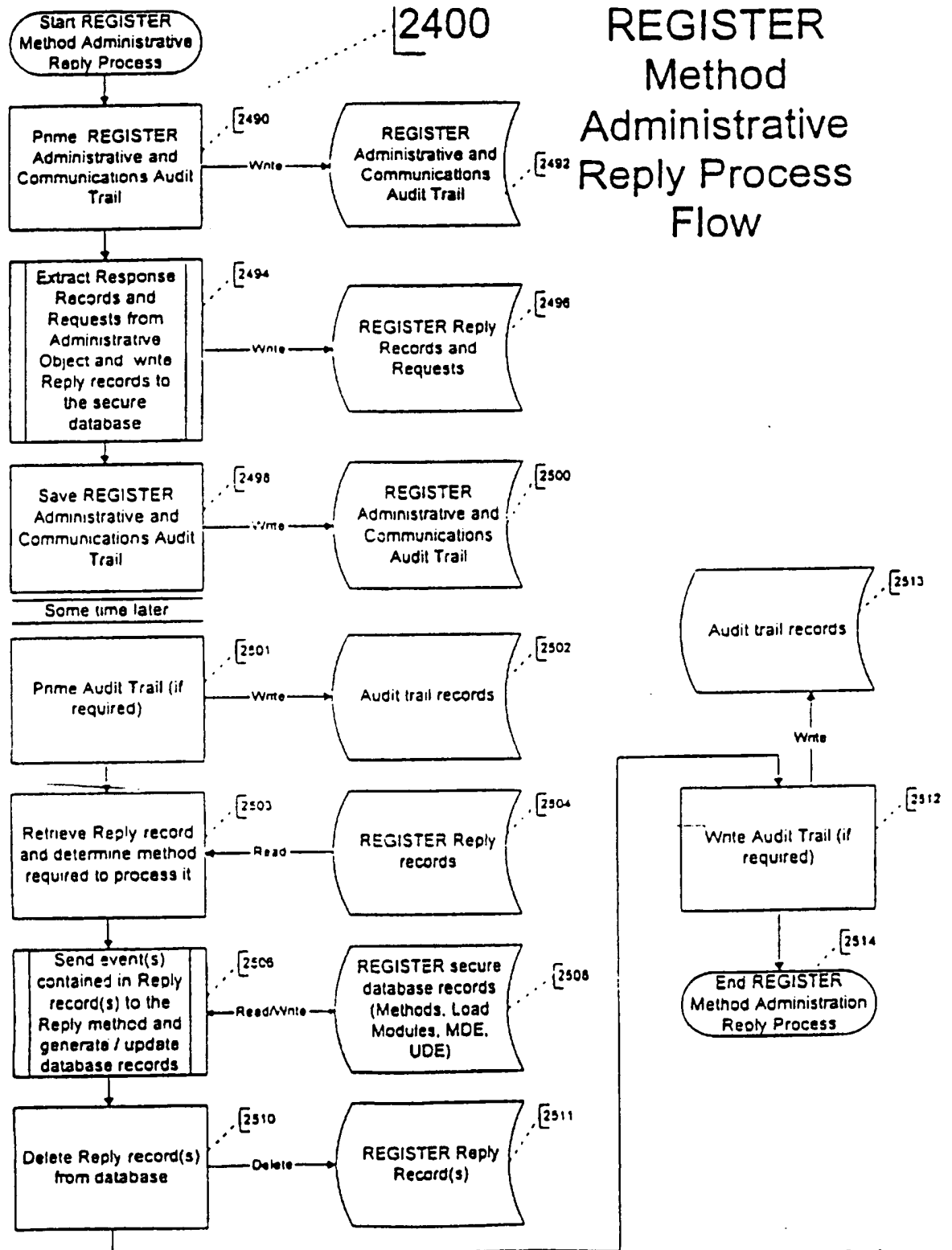


Figure 43d

72/163

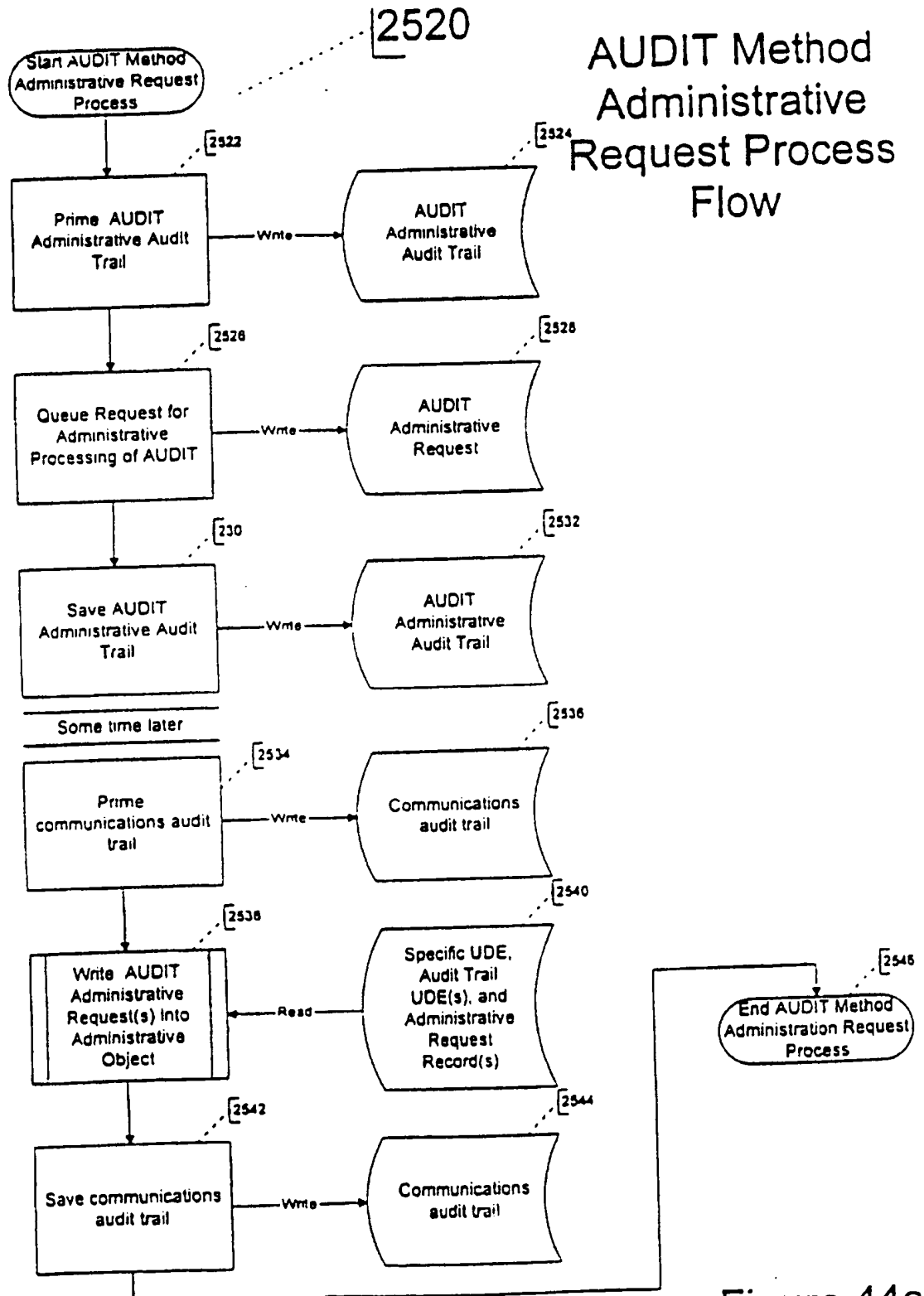


Figure 44a

73/163

# AUDIT Method Administrative Response Process Flow

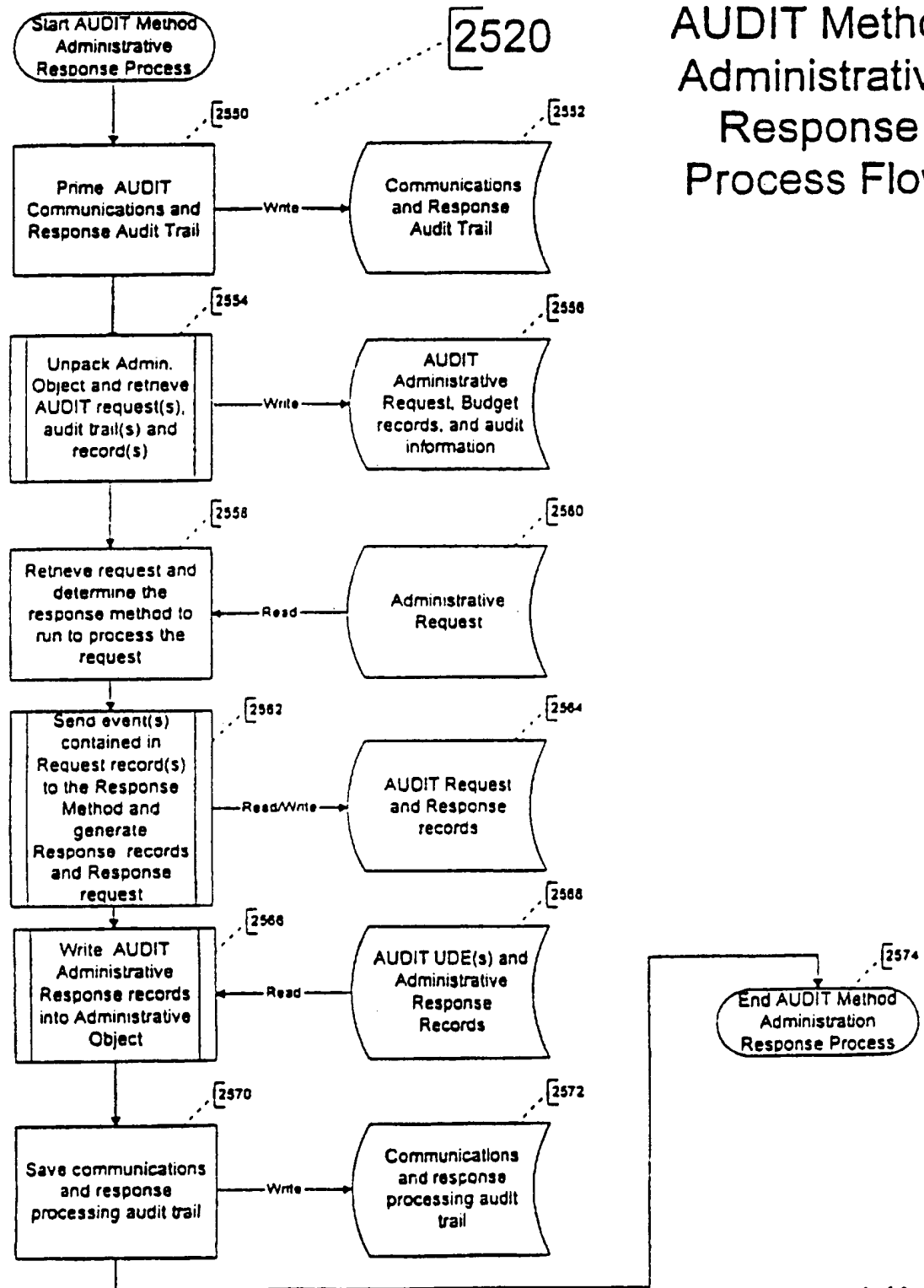


Figure 44b

74/163

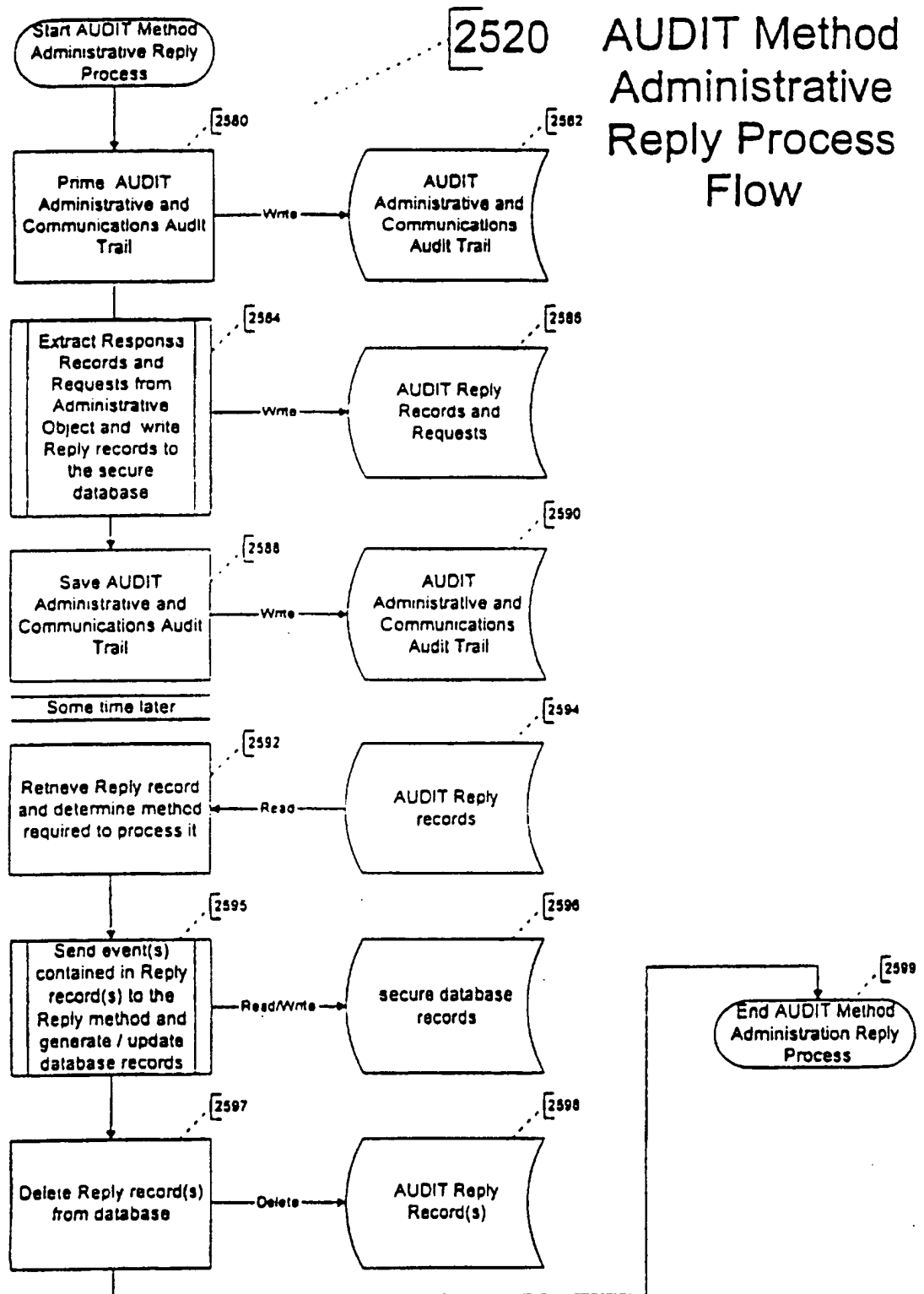
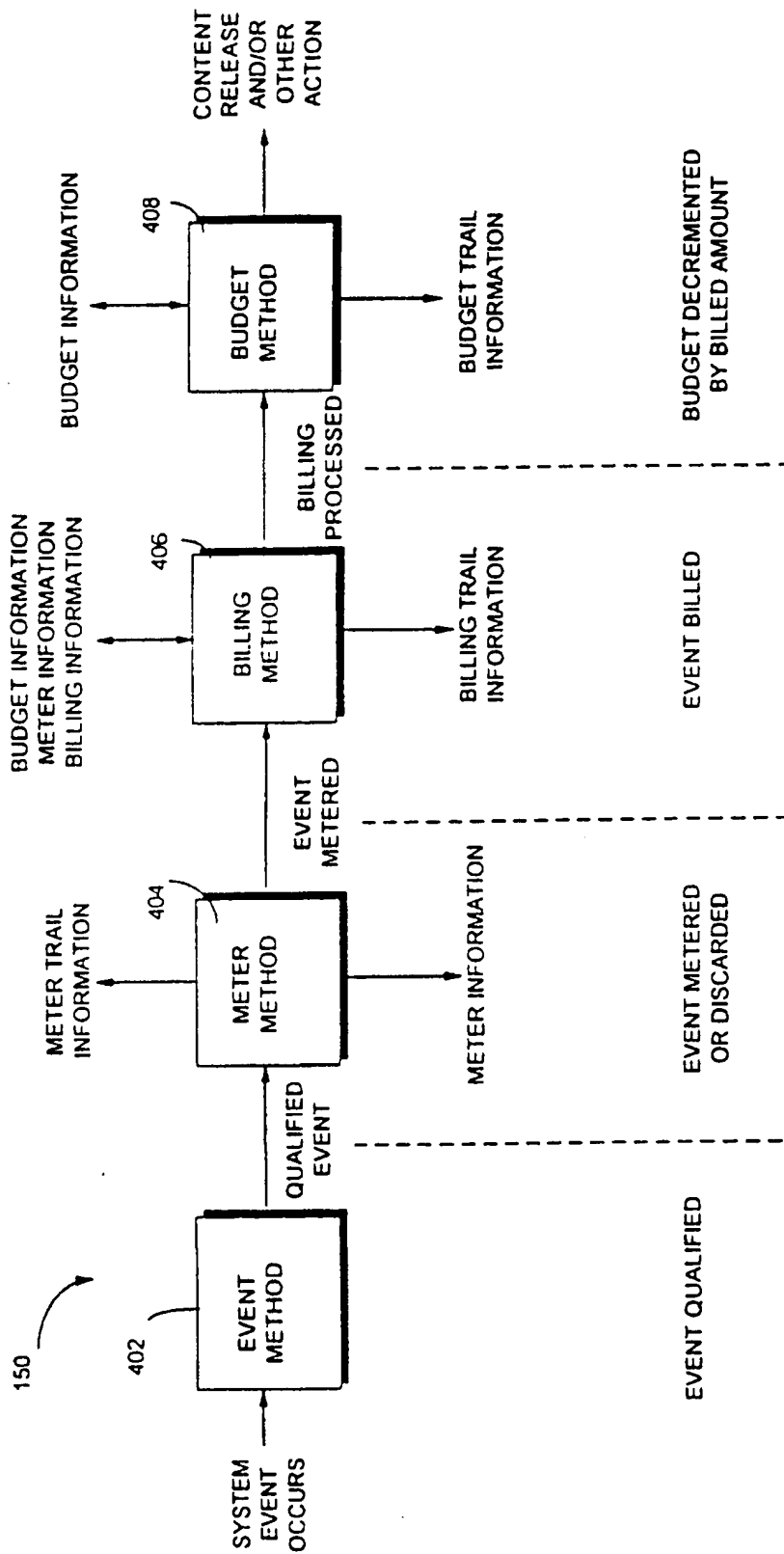


Figure 44c

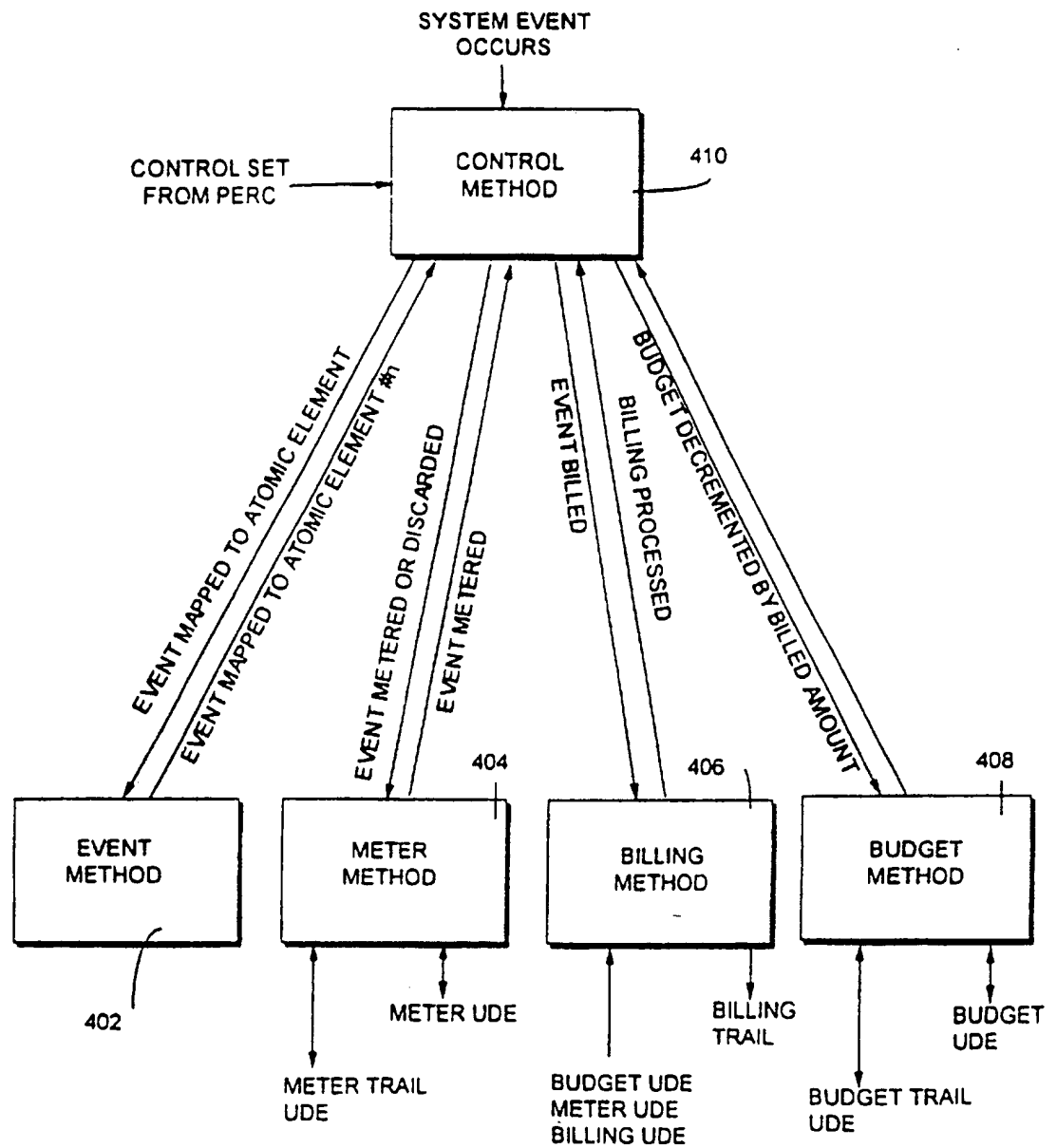
75/163

FIG. 45



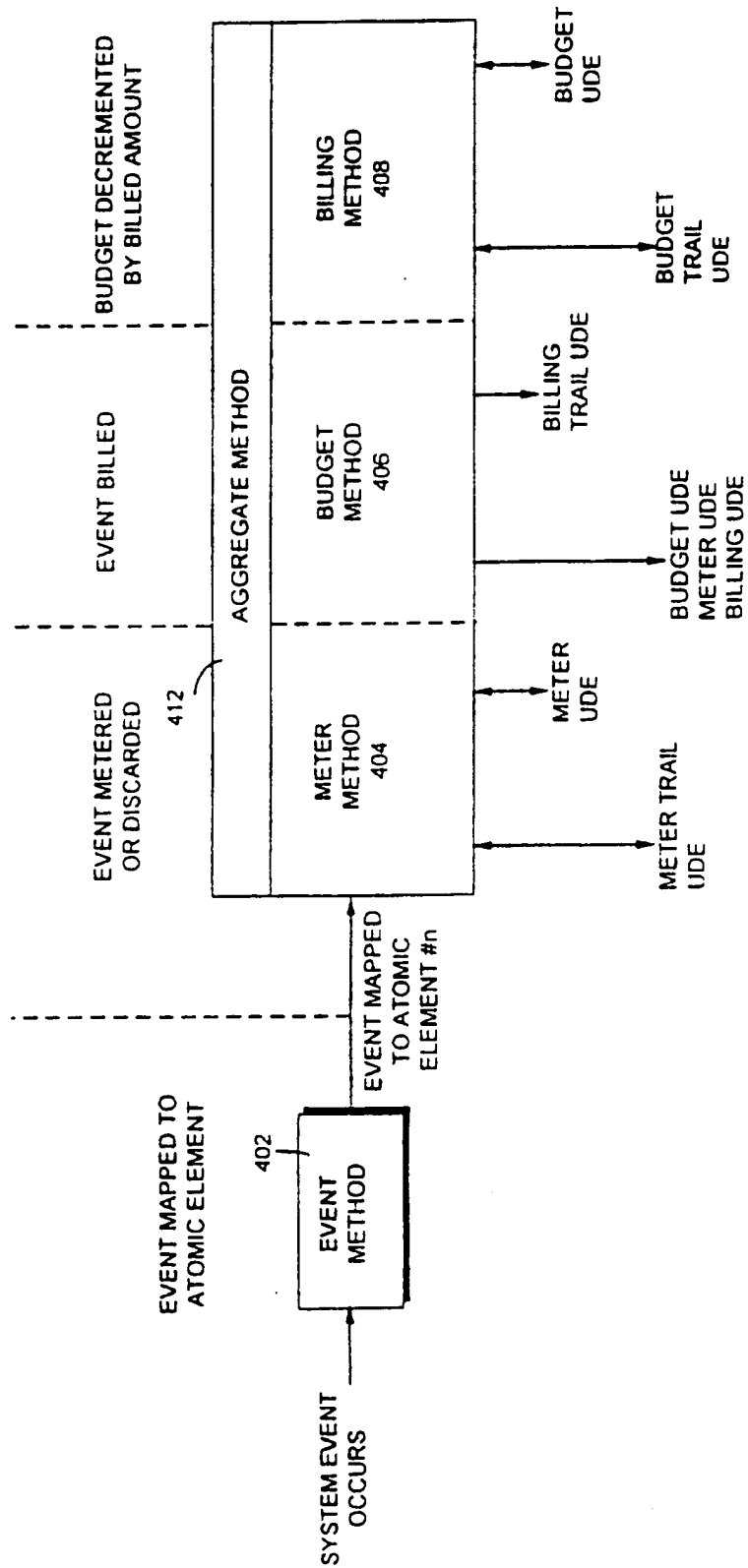
76/163

FIG. 46



77/163

FIG. 47



78/163

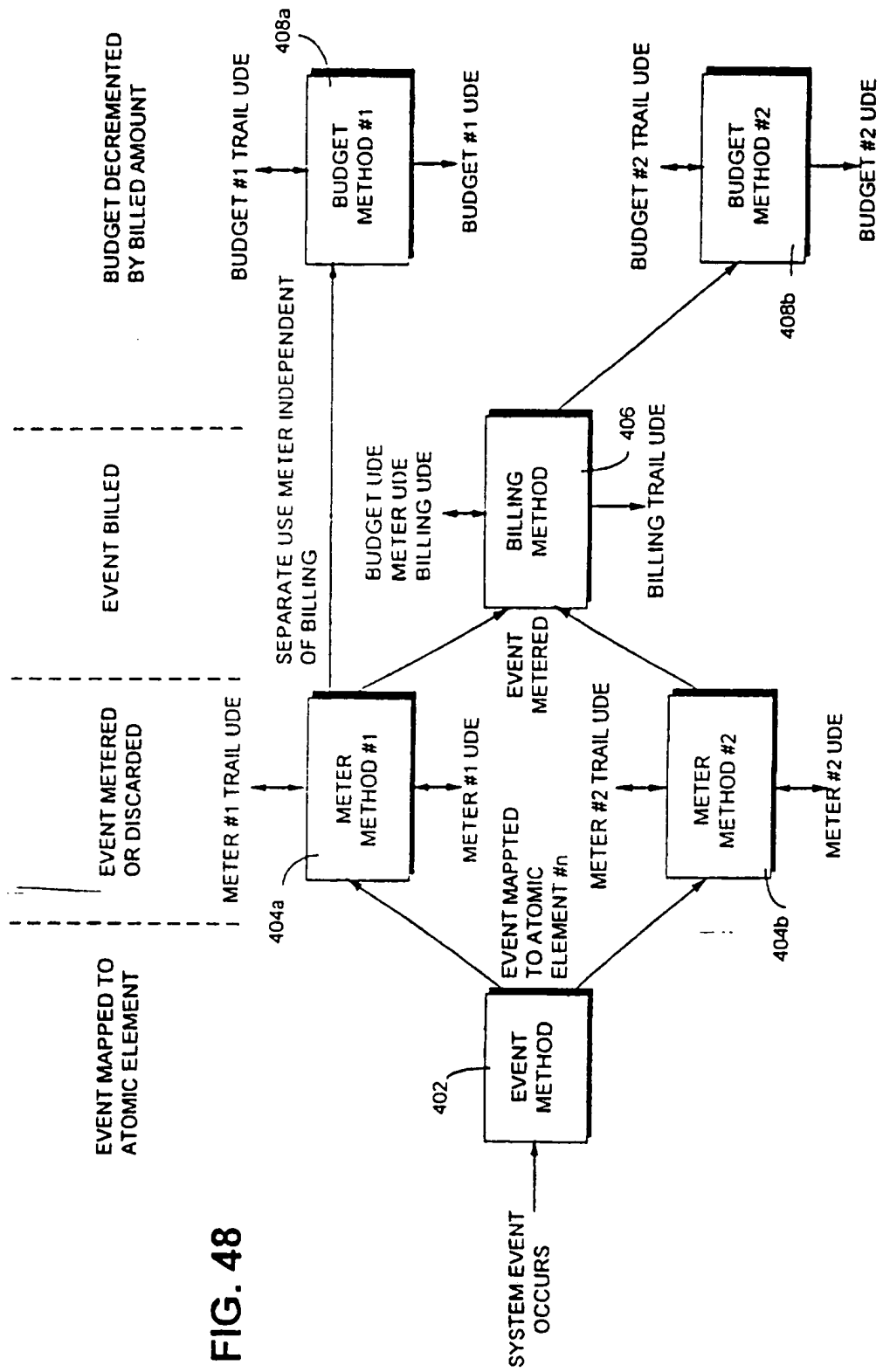


FIG. 48



79/163

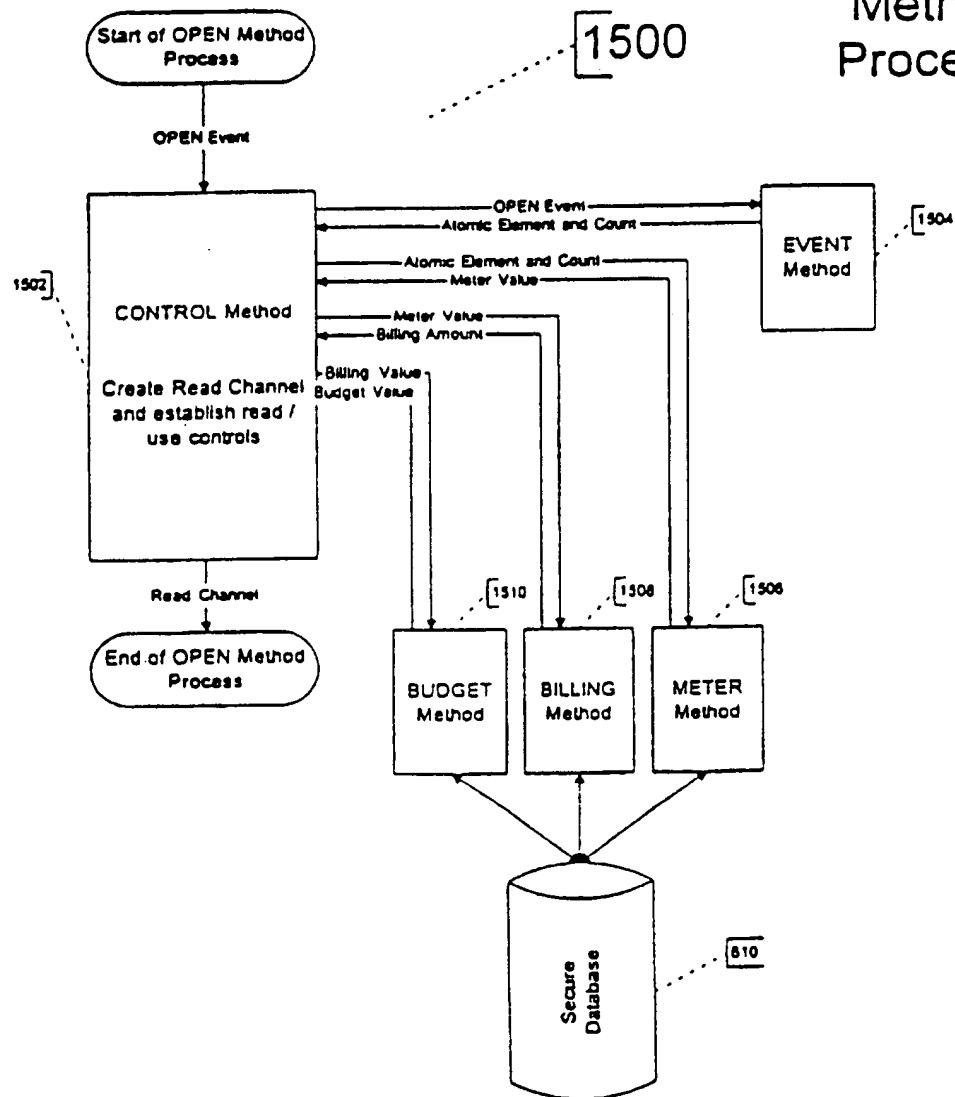
OPEN  
Method Use  
Process Flow

Figure 49

80/163

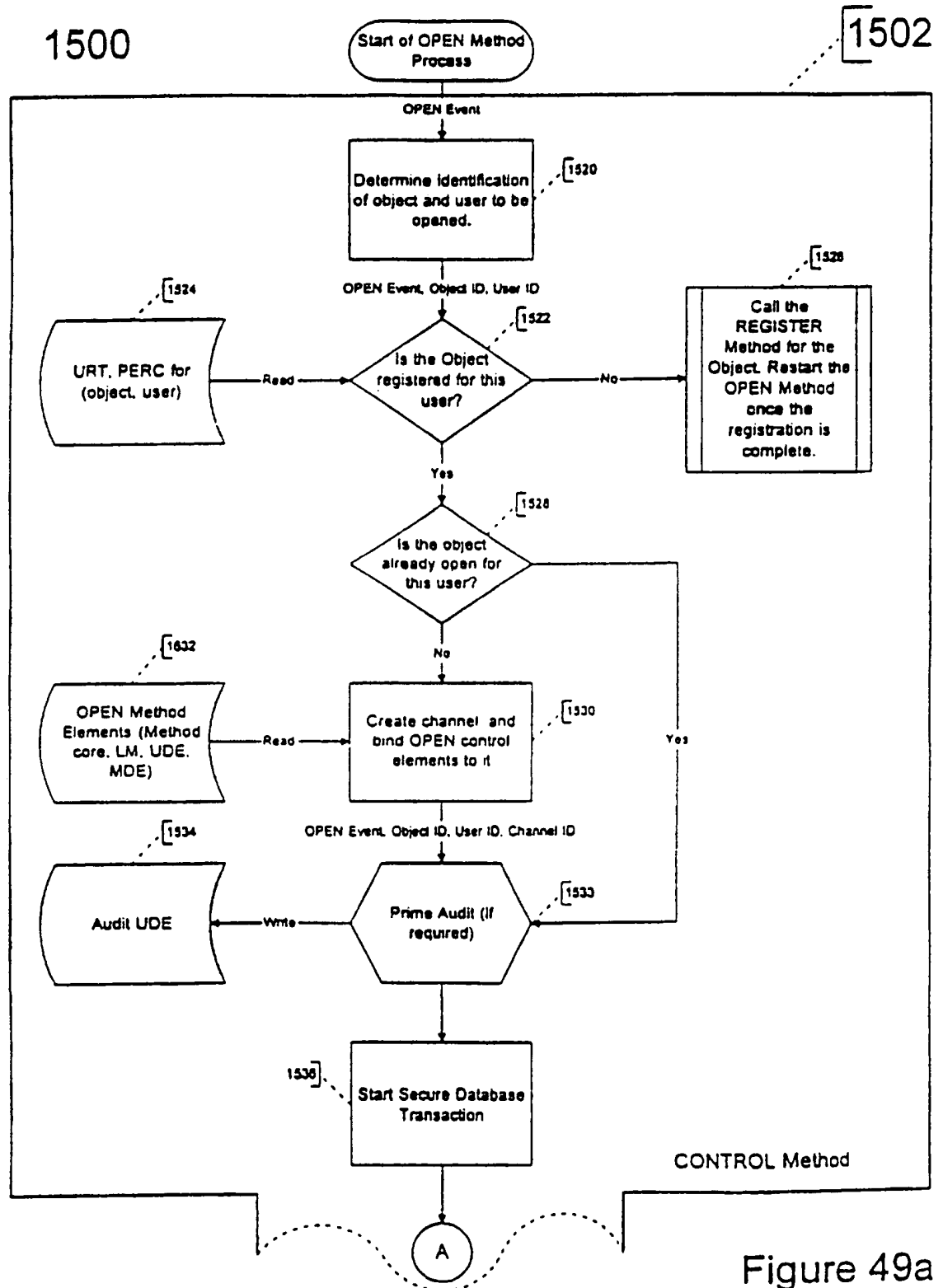
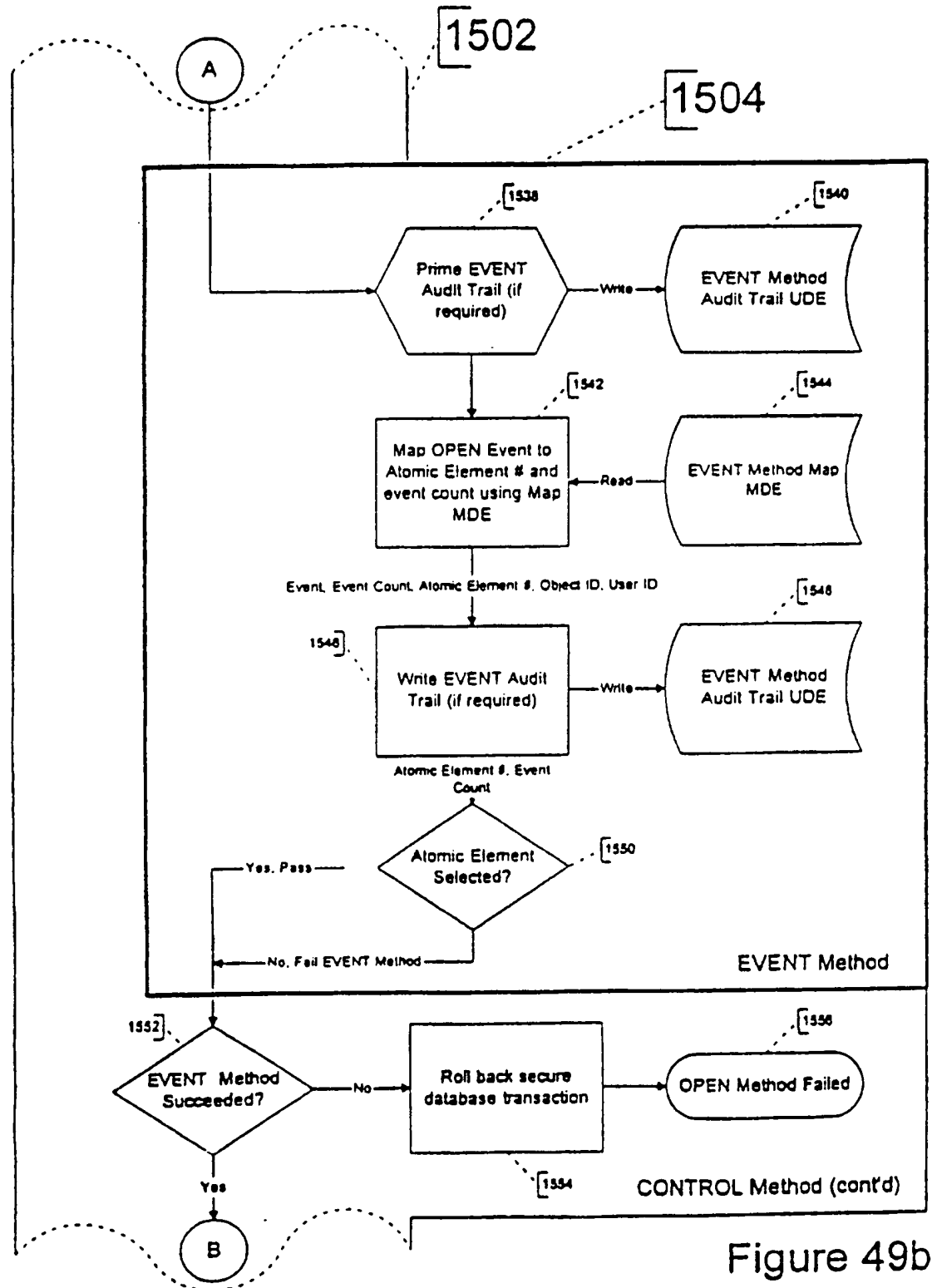
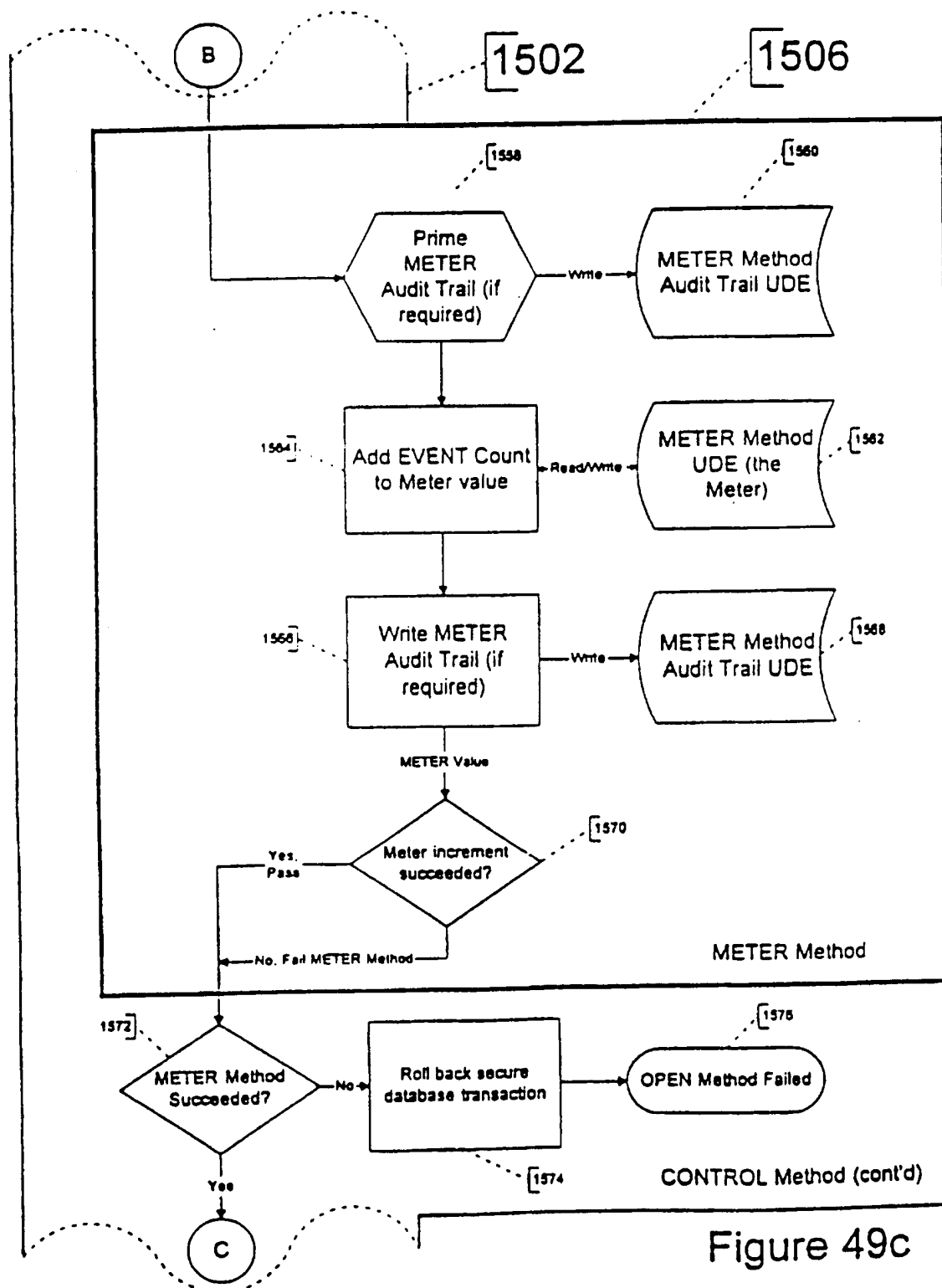


Figure 49a

81/163



82/163



83/163

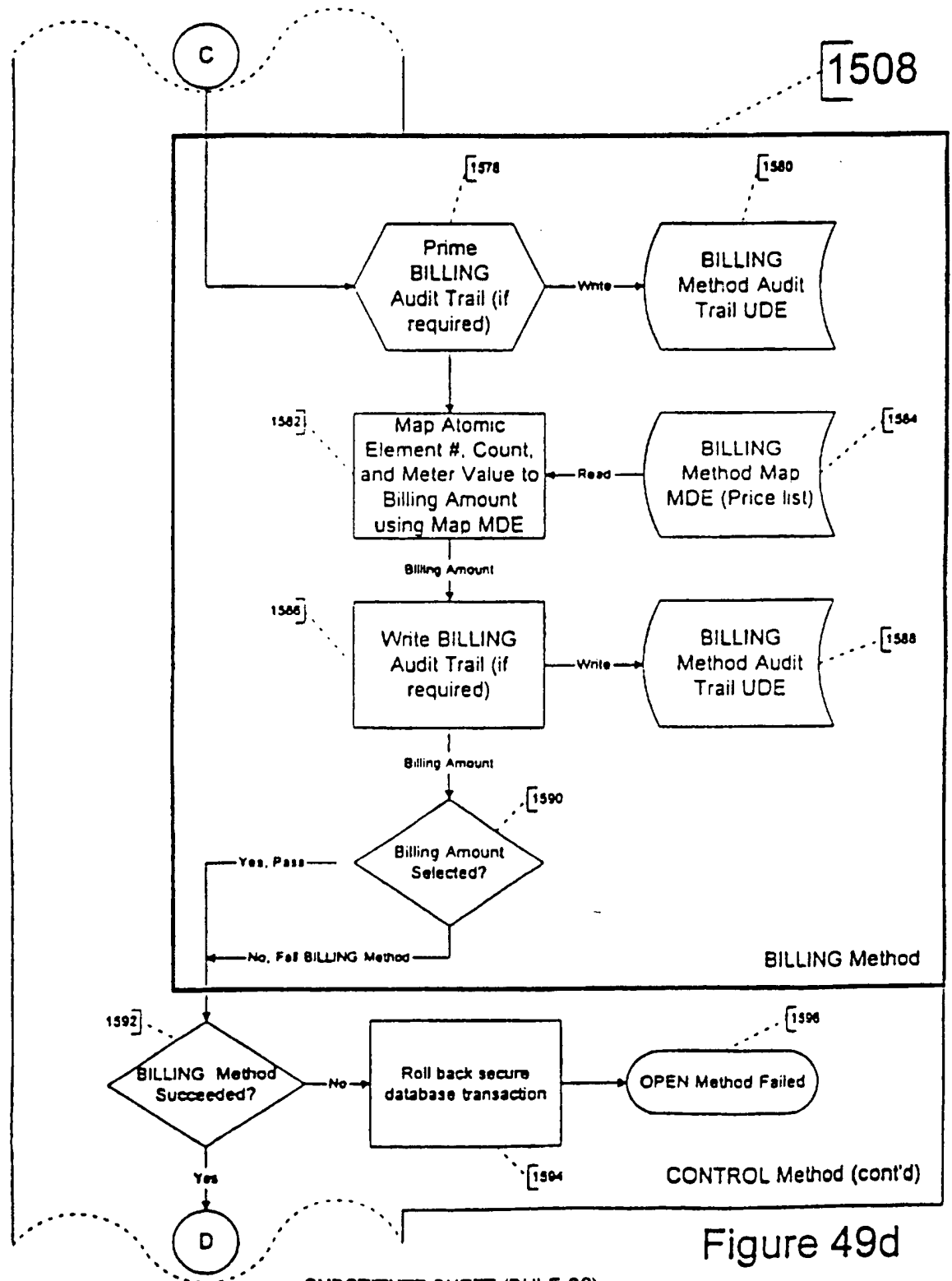
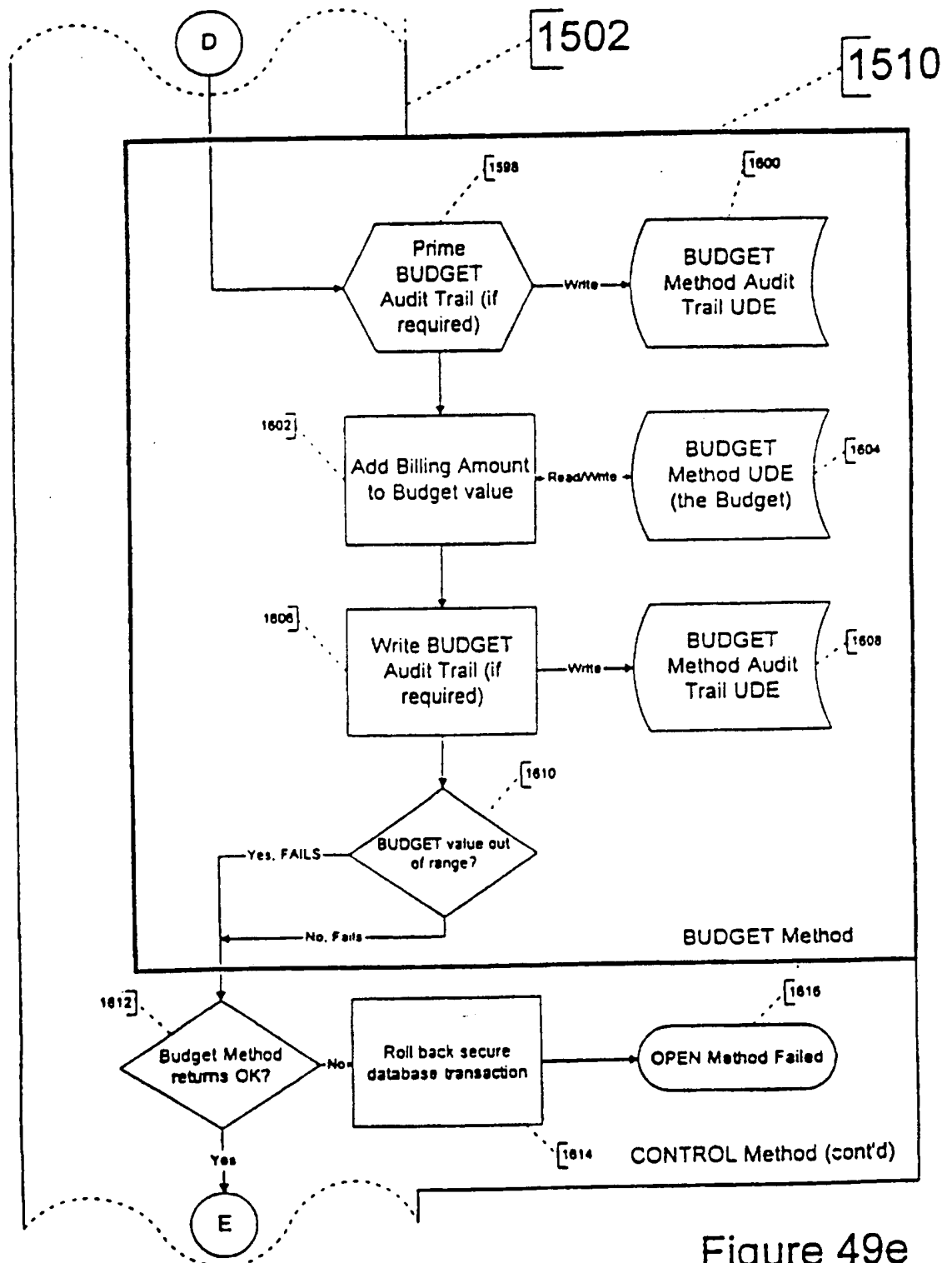


Figure 49d

84/163



SUBSTITUTE SHEET (RULE 26)

Figure 49e

85/163

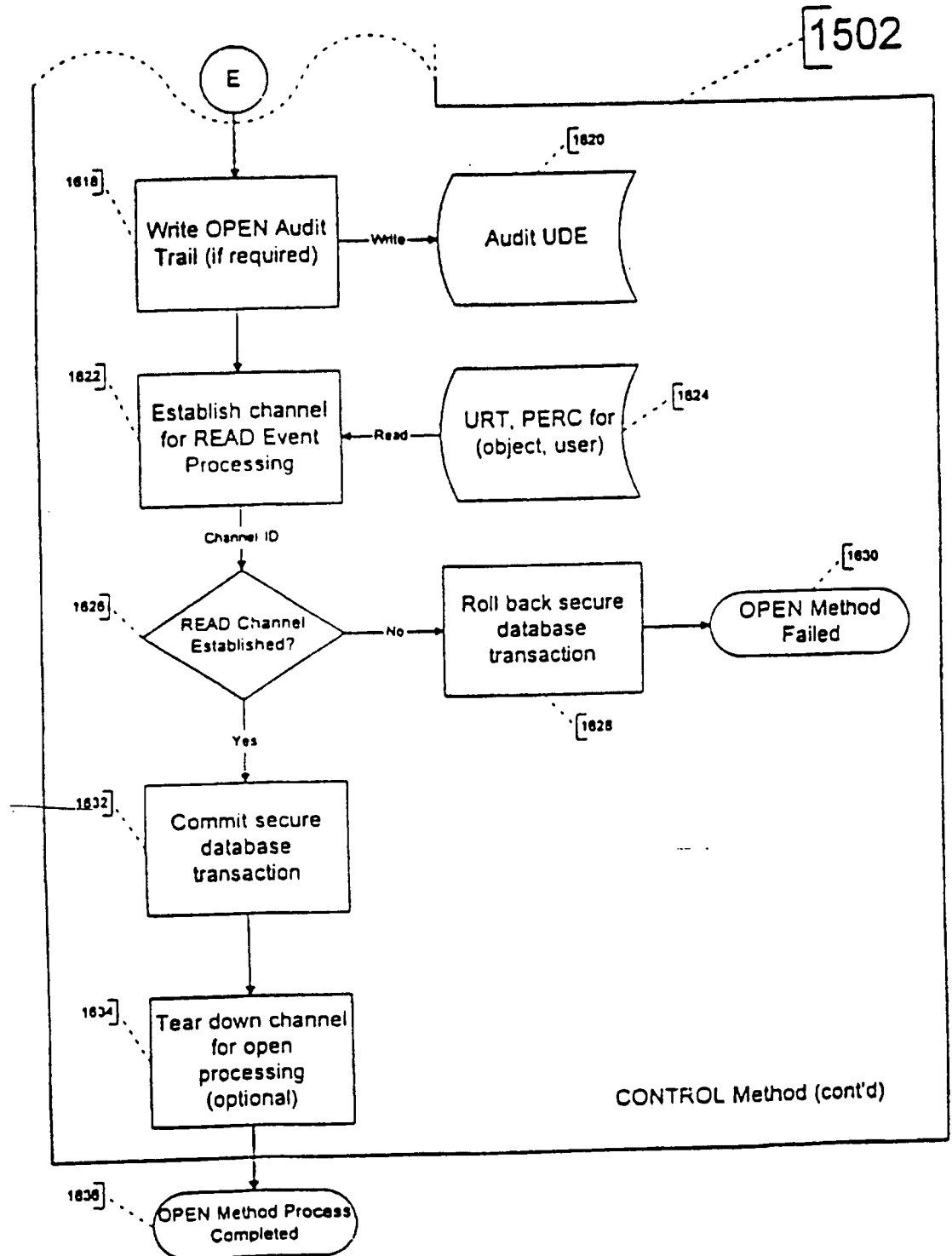


Figure 49f

86/163

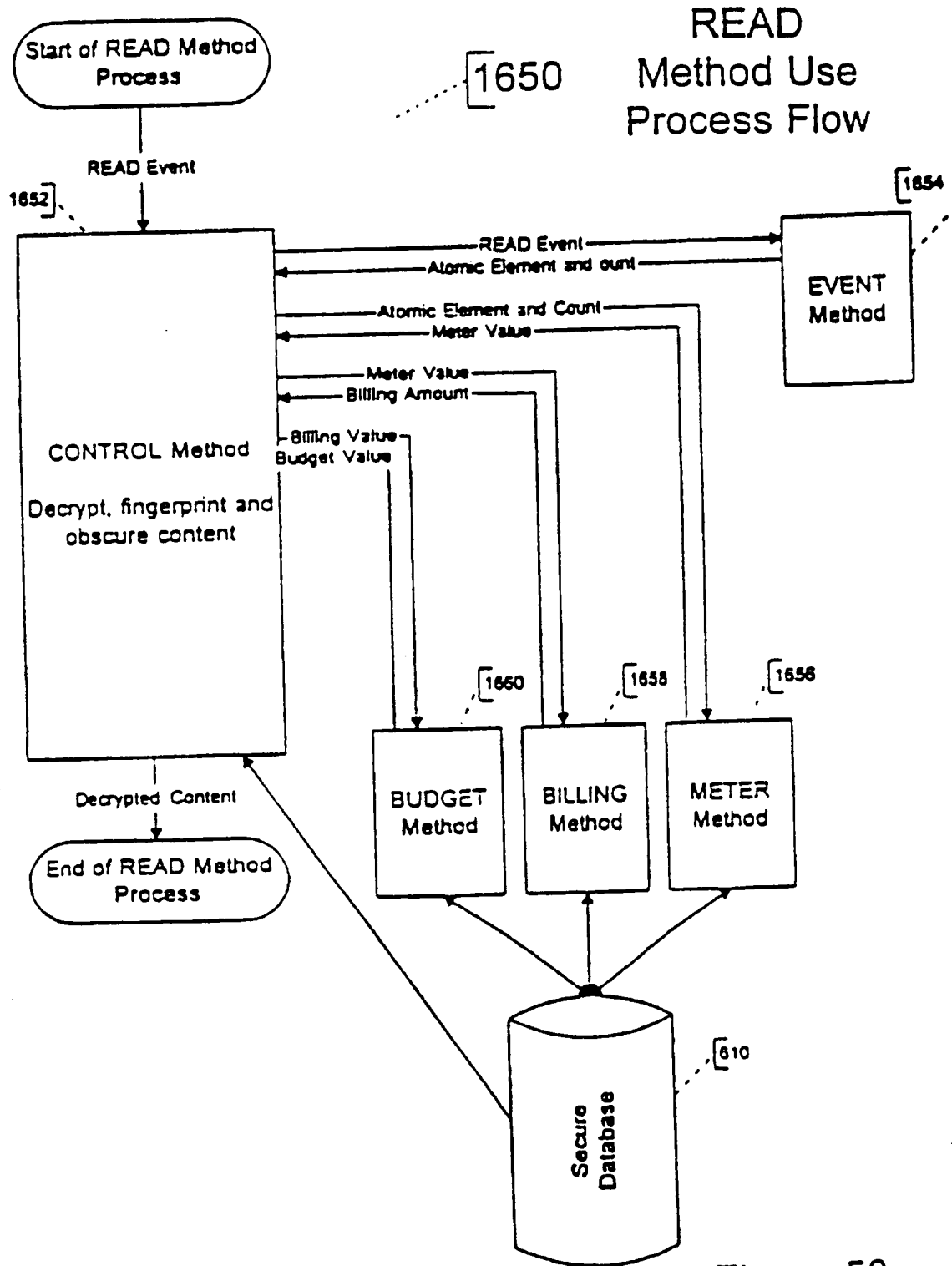


Figure 50



87/163

1650

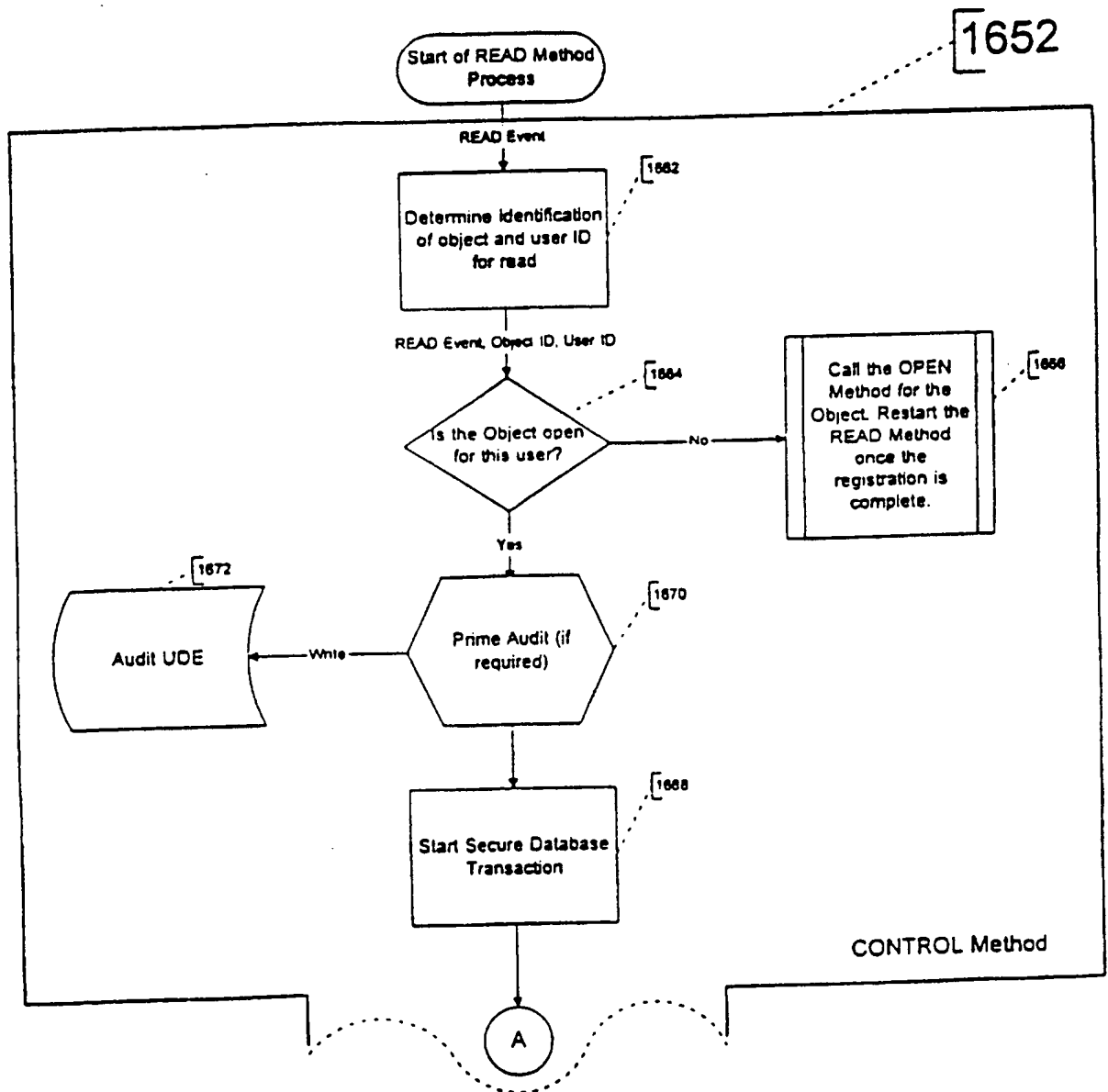


Figure 50a

88/163

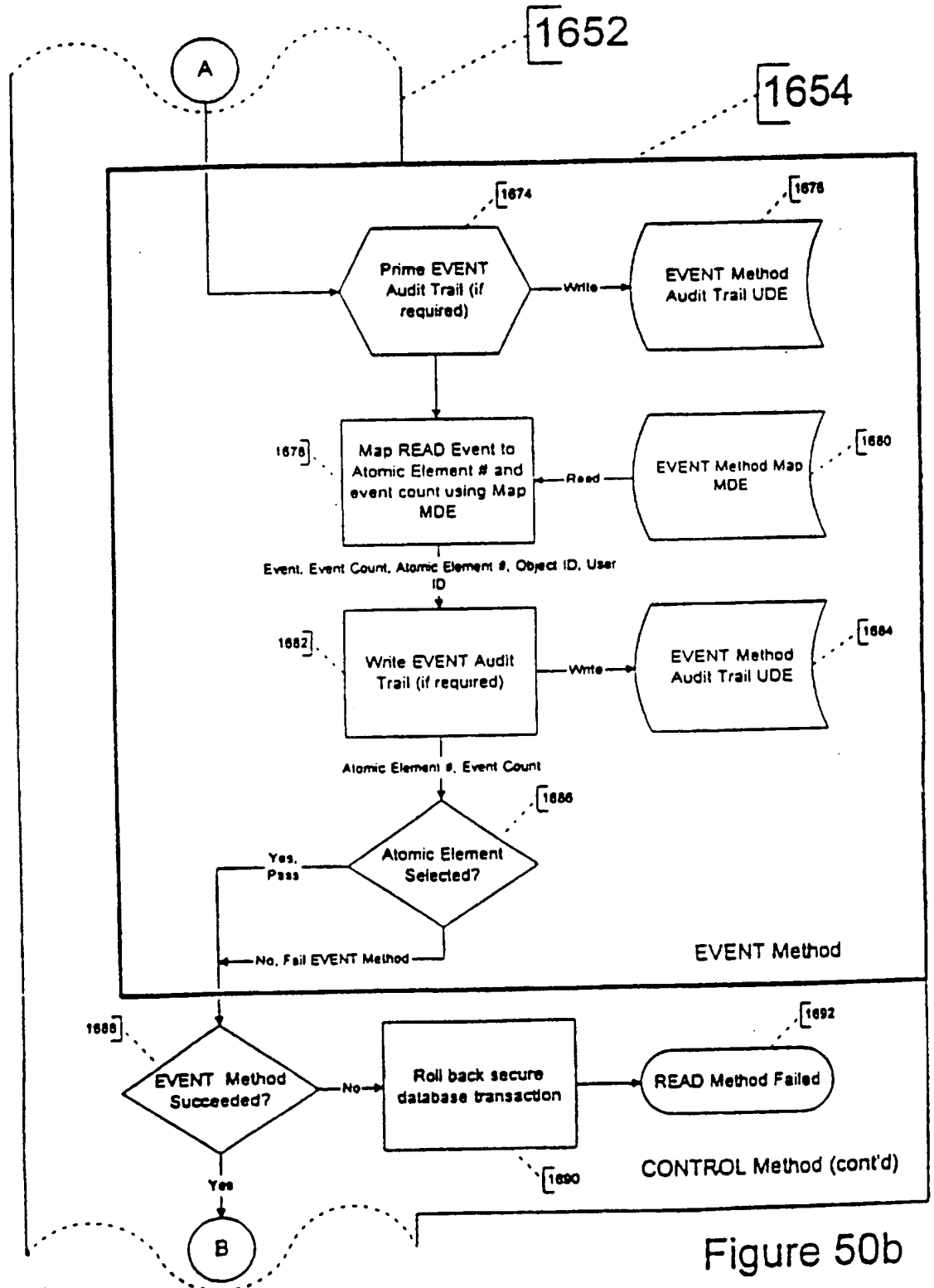
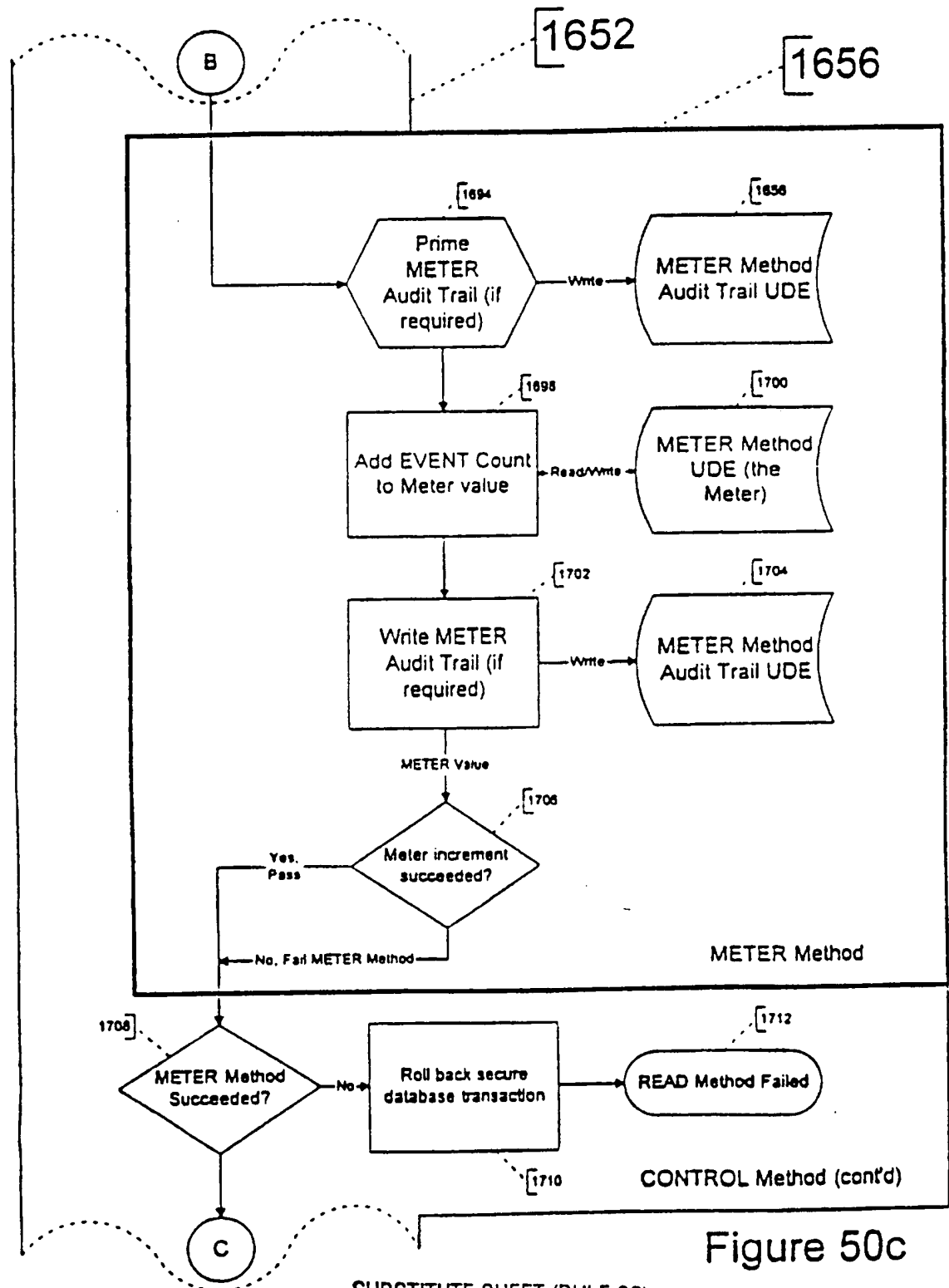


Figure 50b

89/163



90/163

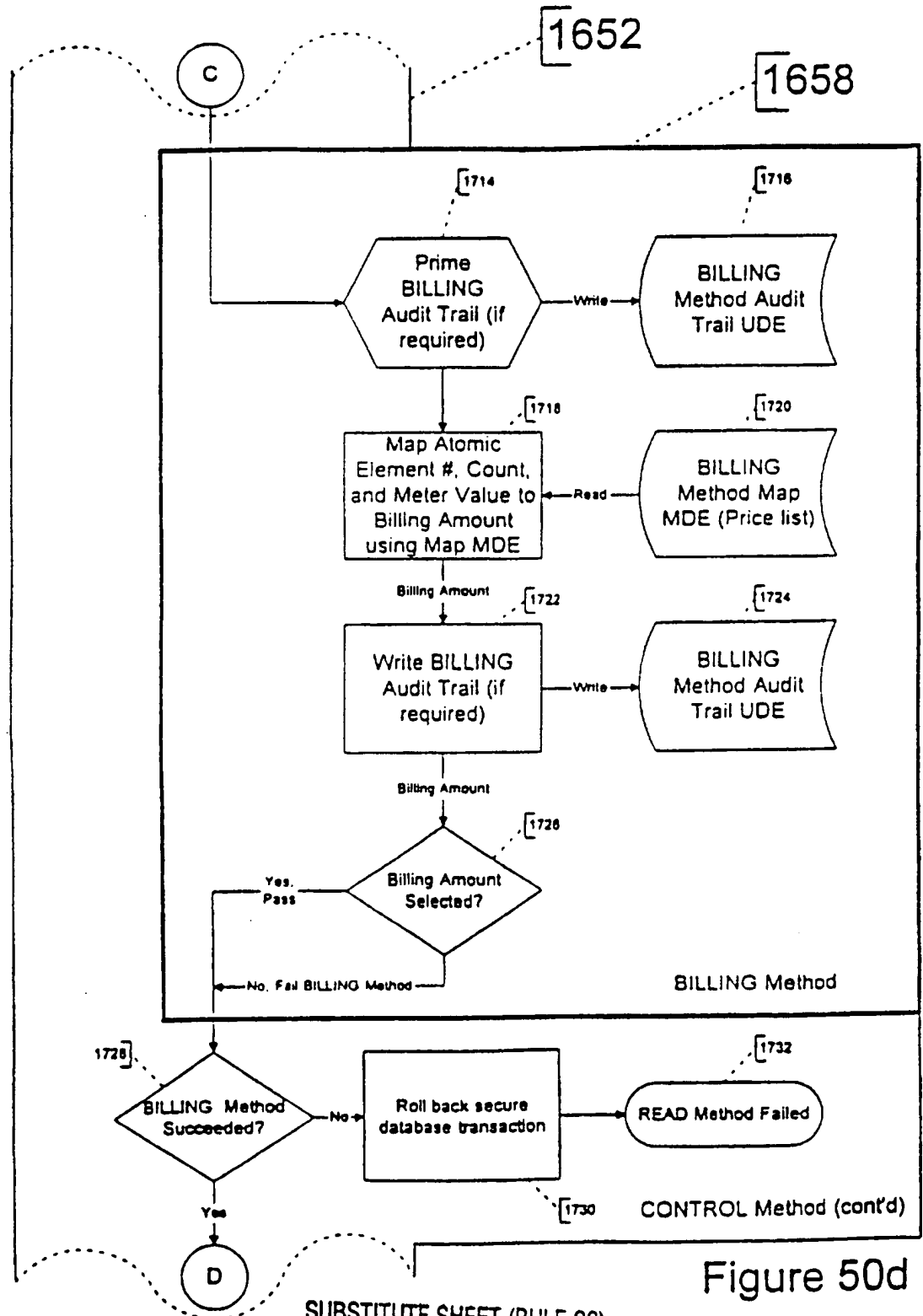


Figure 50d

91/163

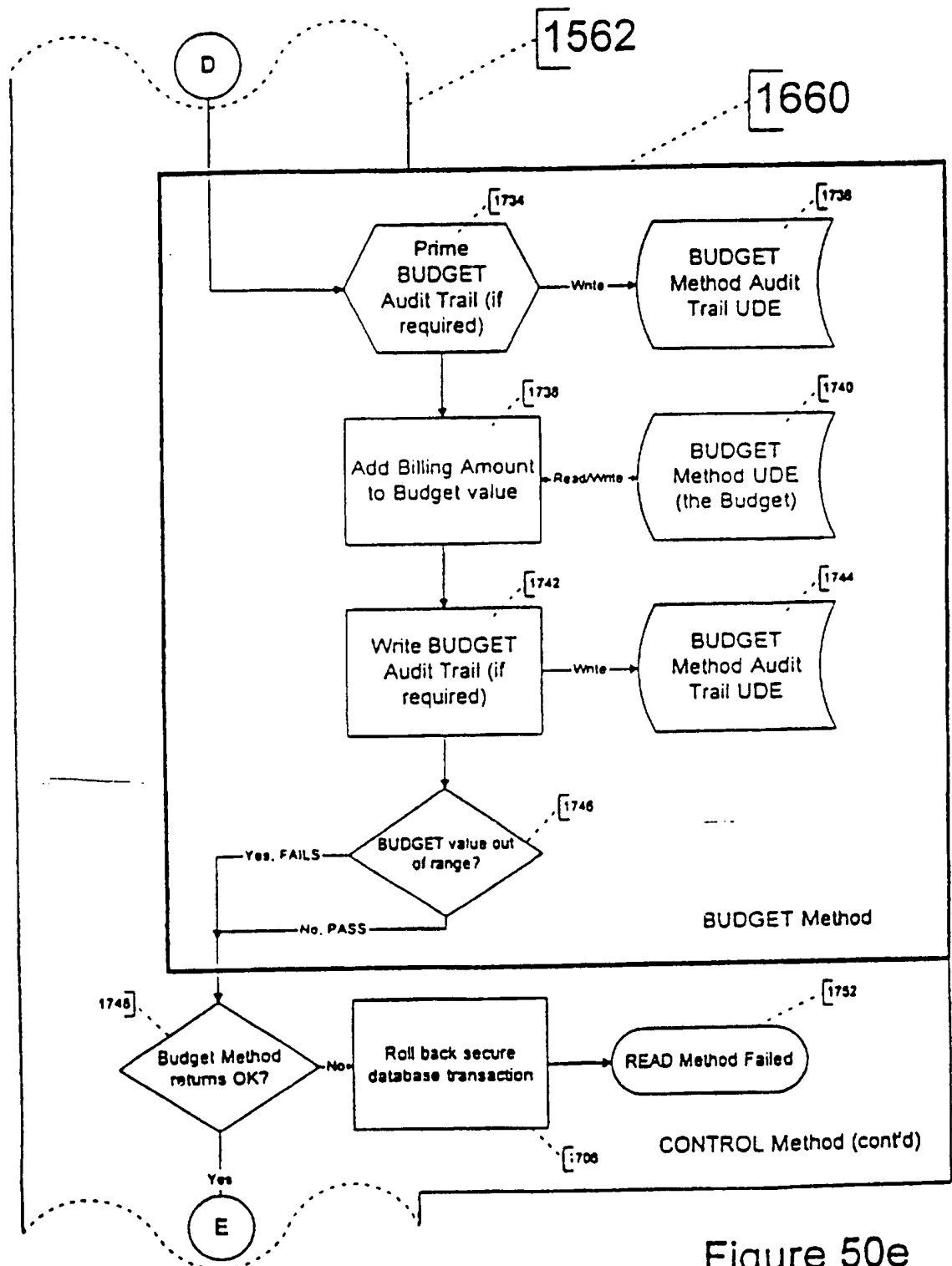
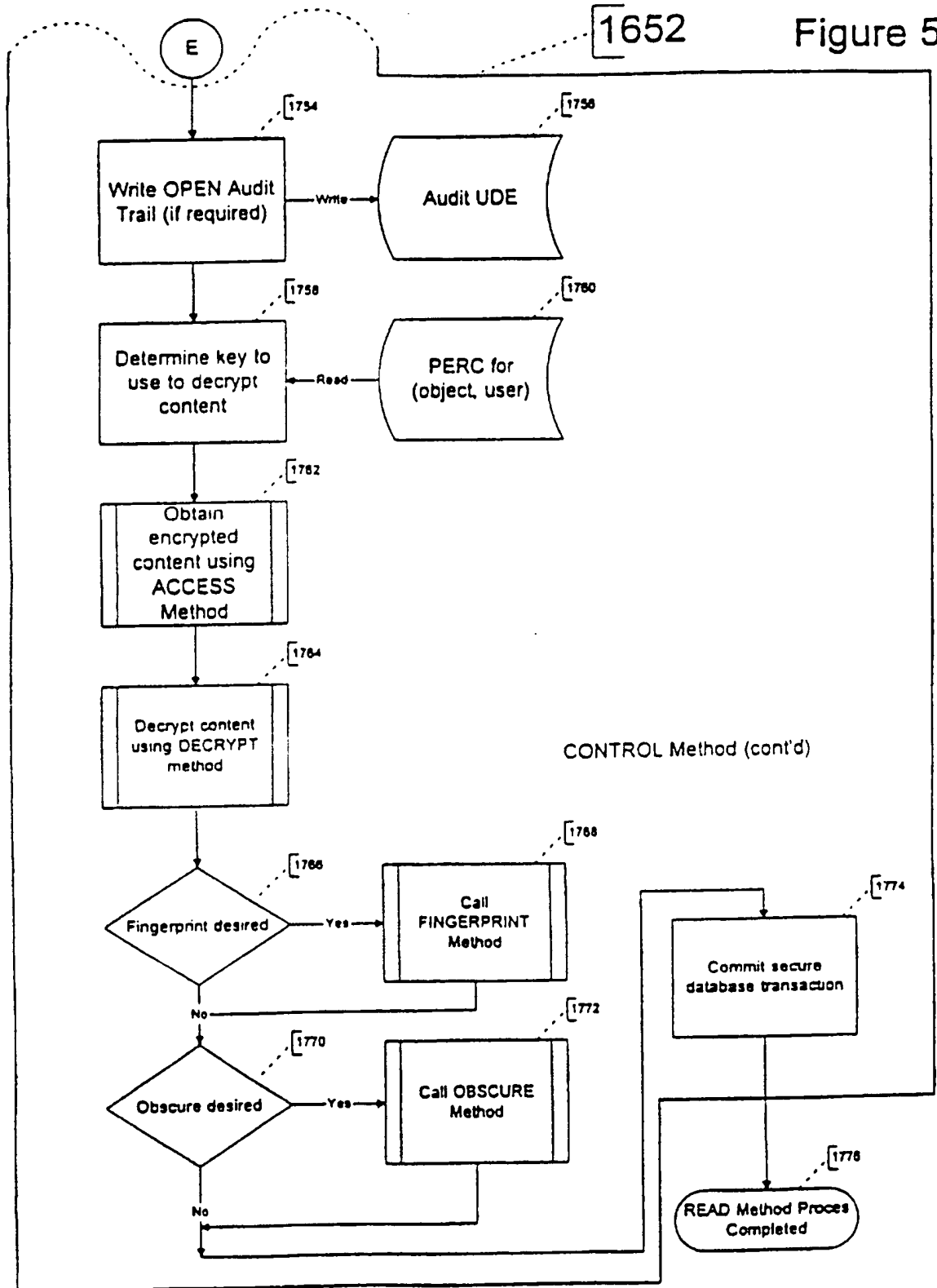


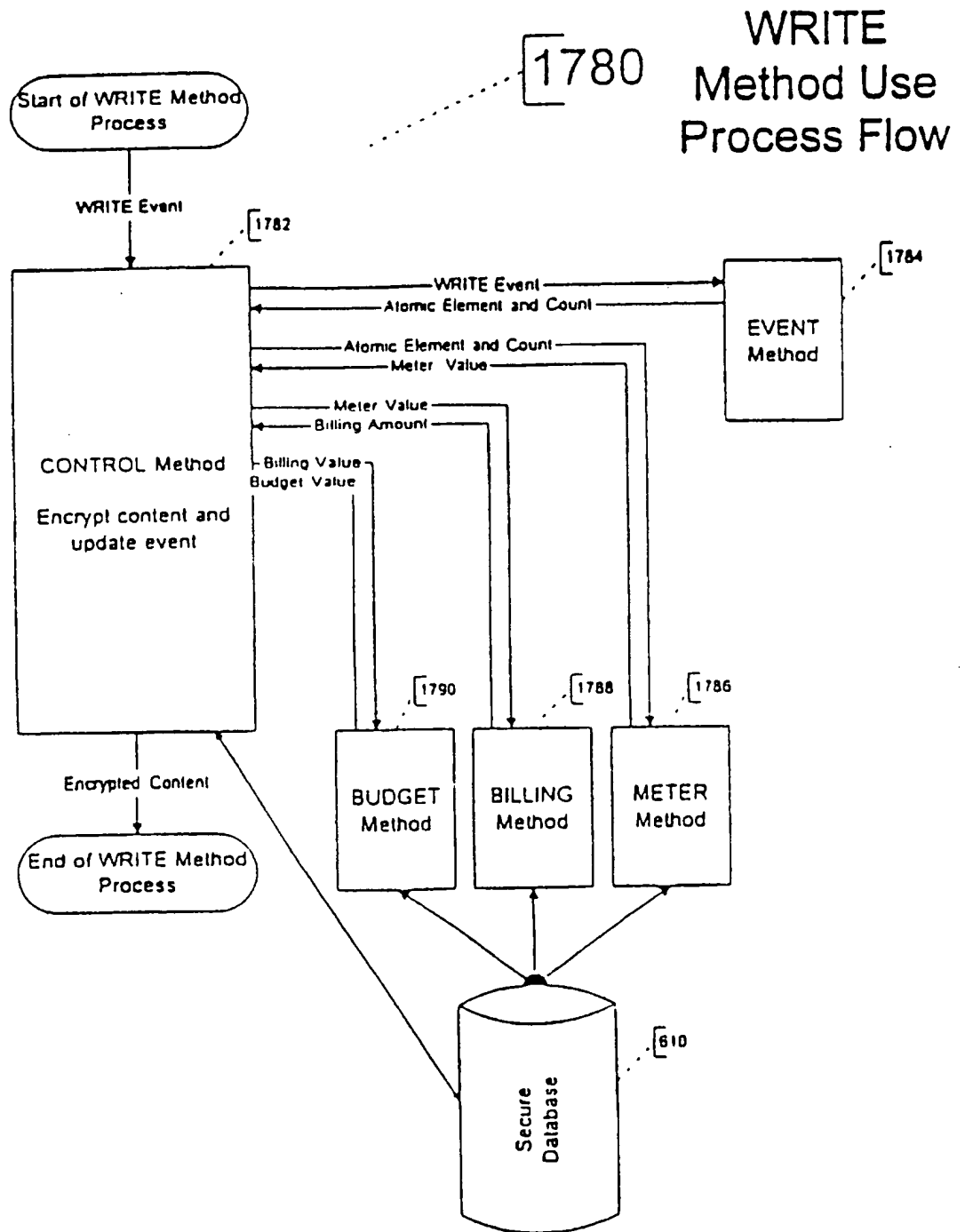
Figure 50e

92/163

Figure 50f

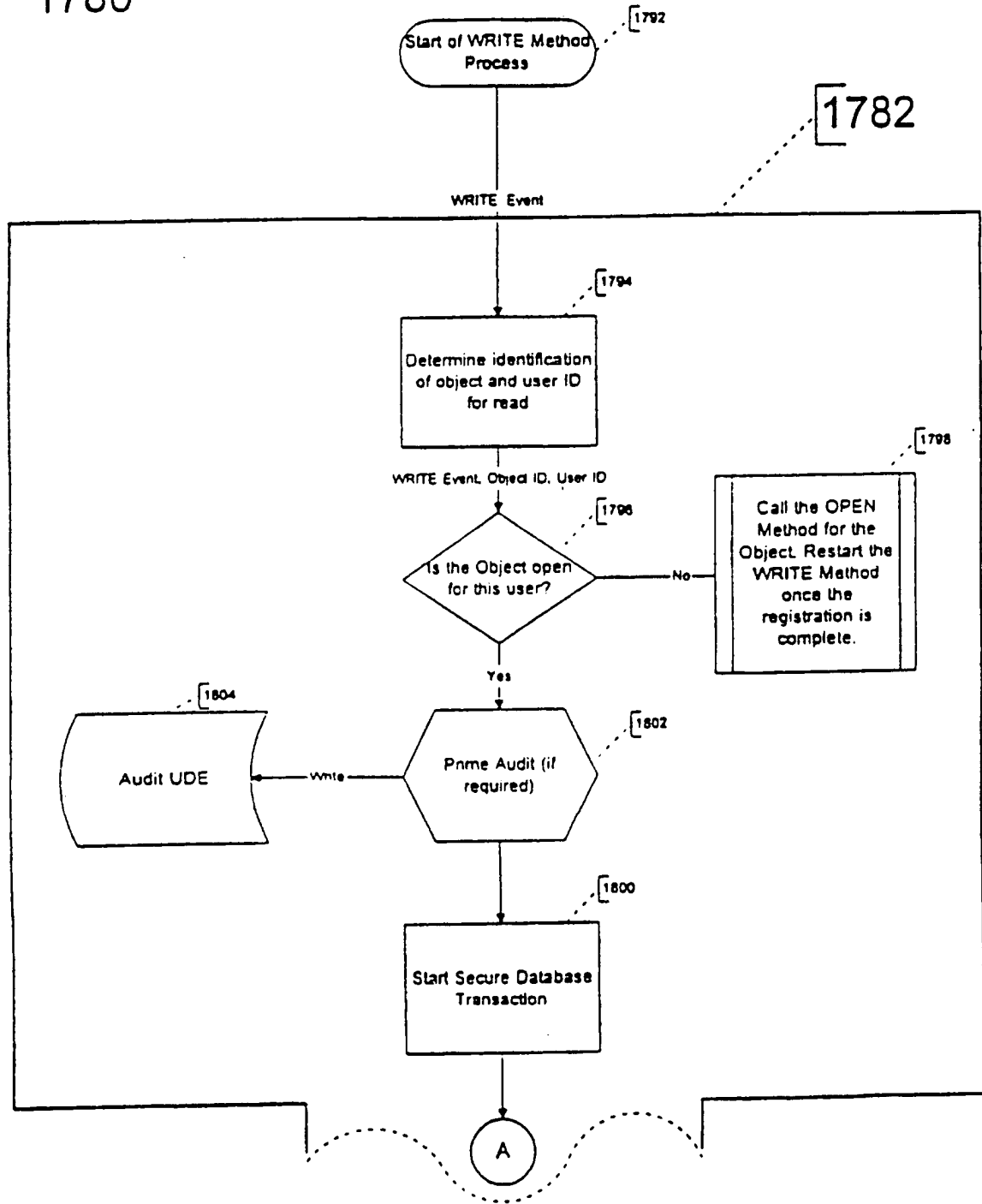


93/163



94/163

1780





95/163

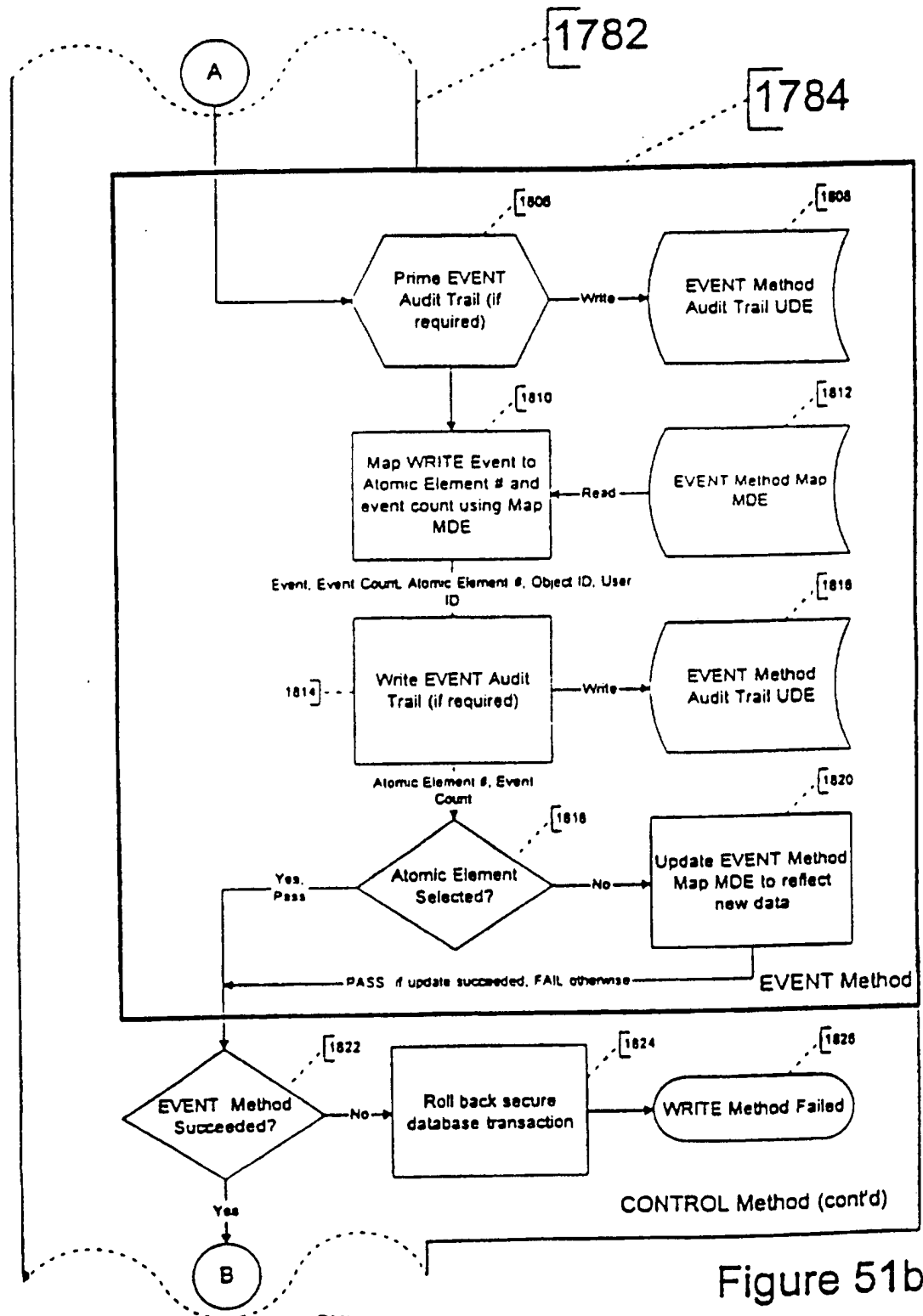
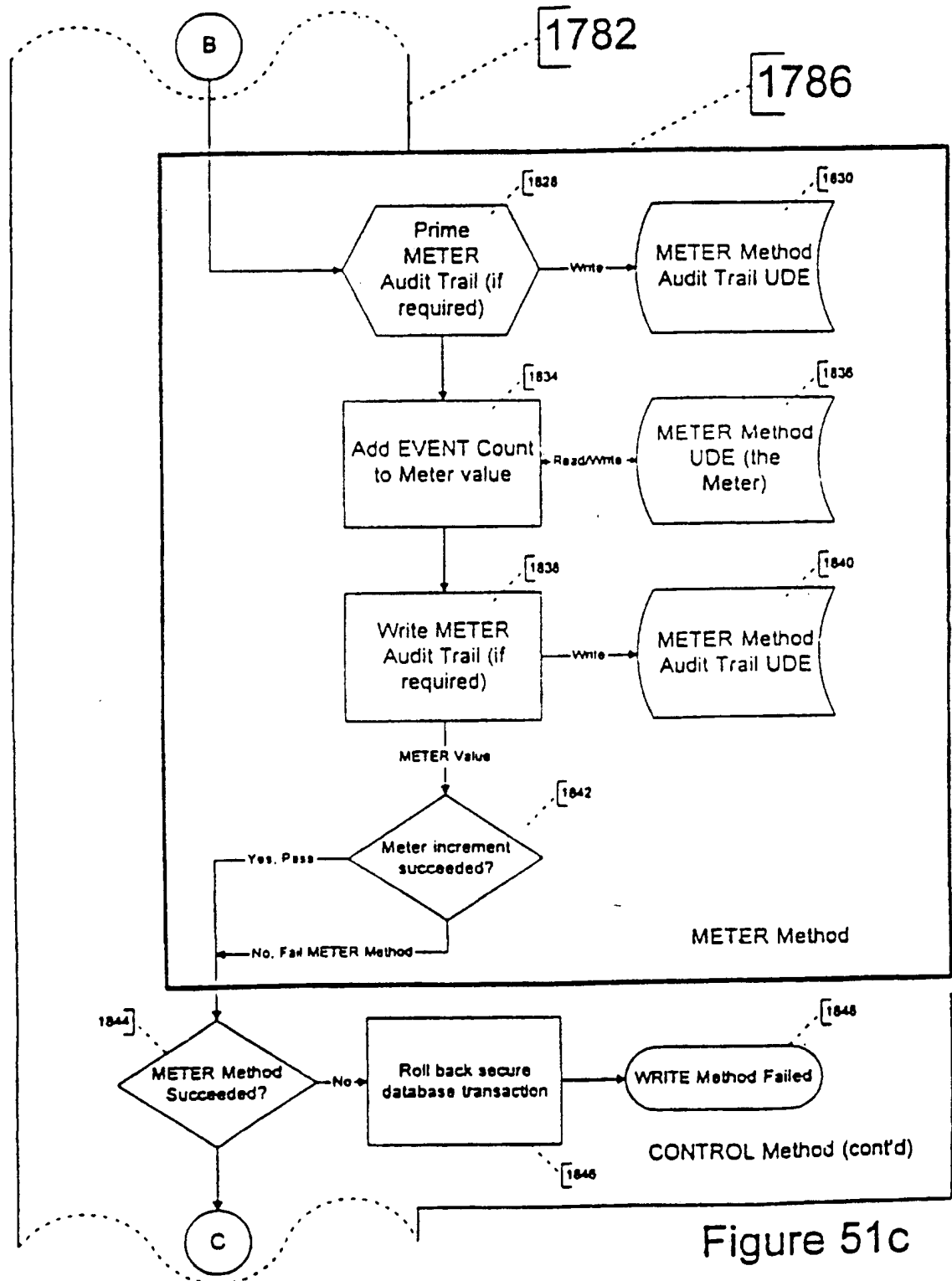


Figure 51b

96/163



97/163

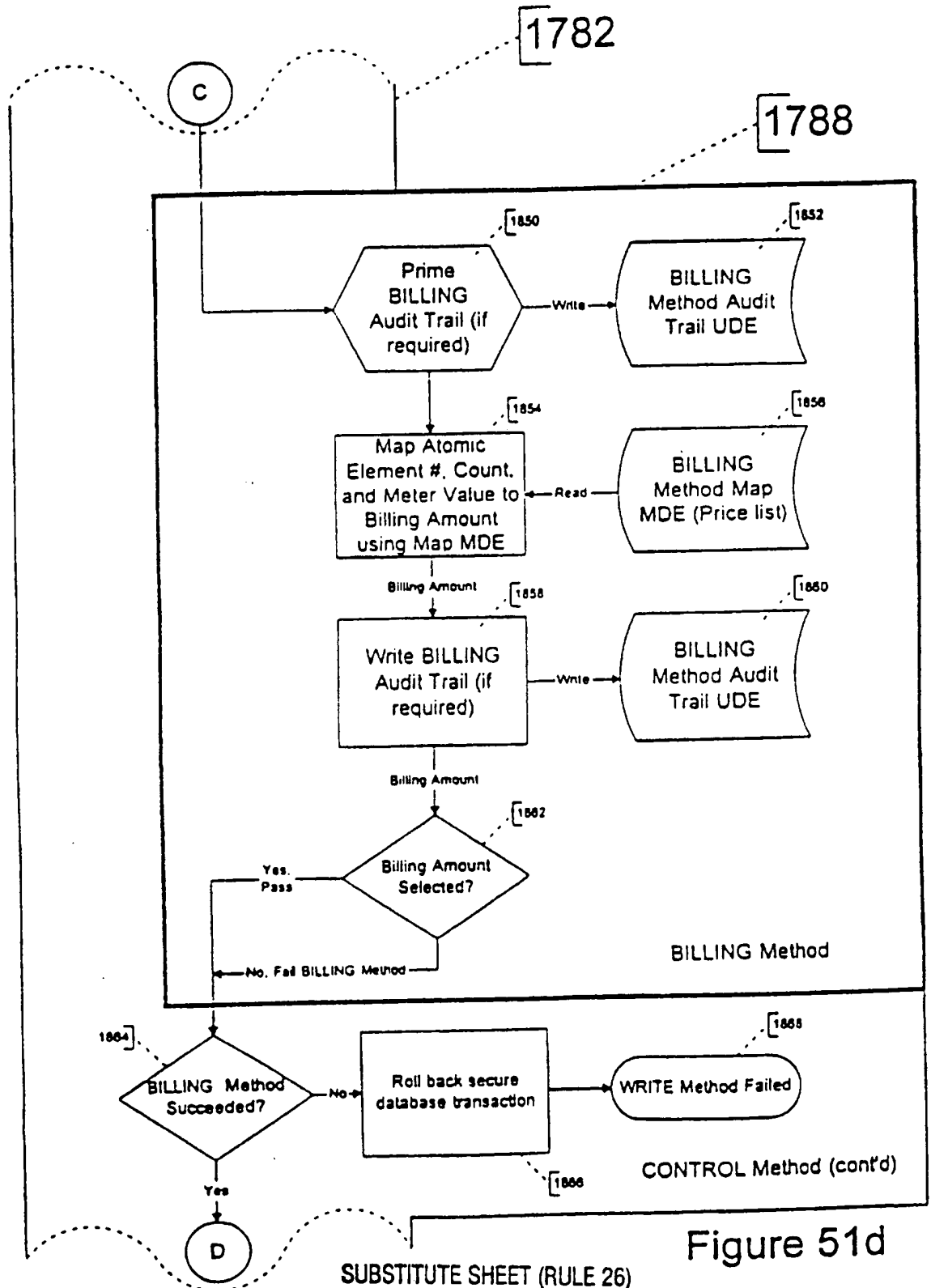
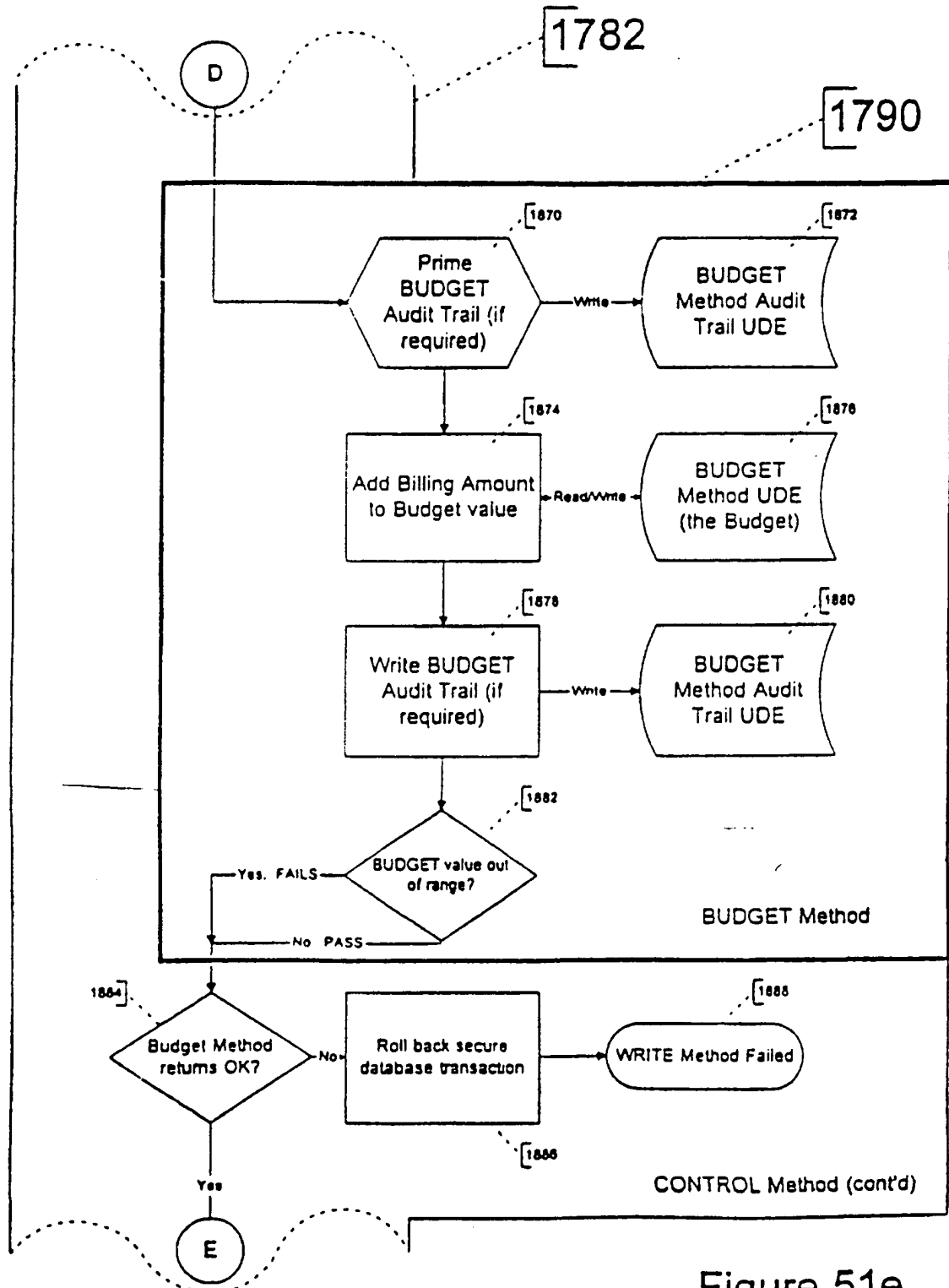


Figure 51d

98/163



99/163

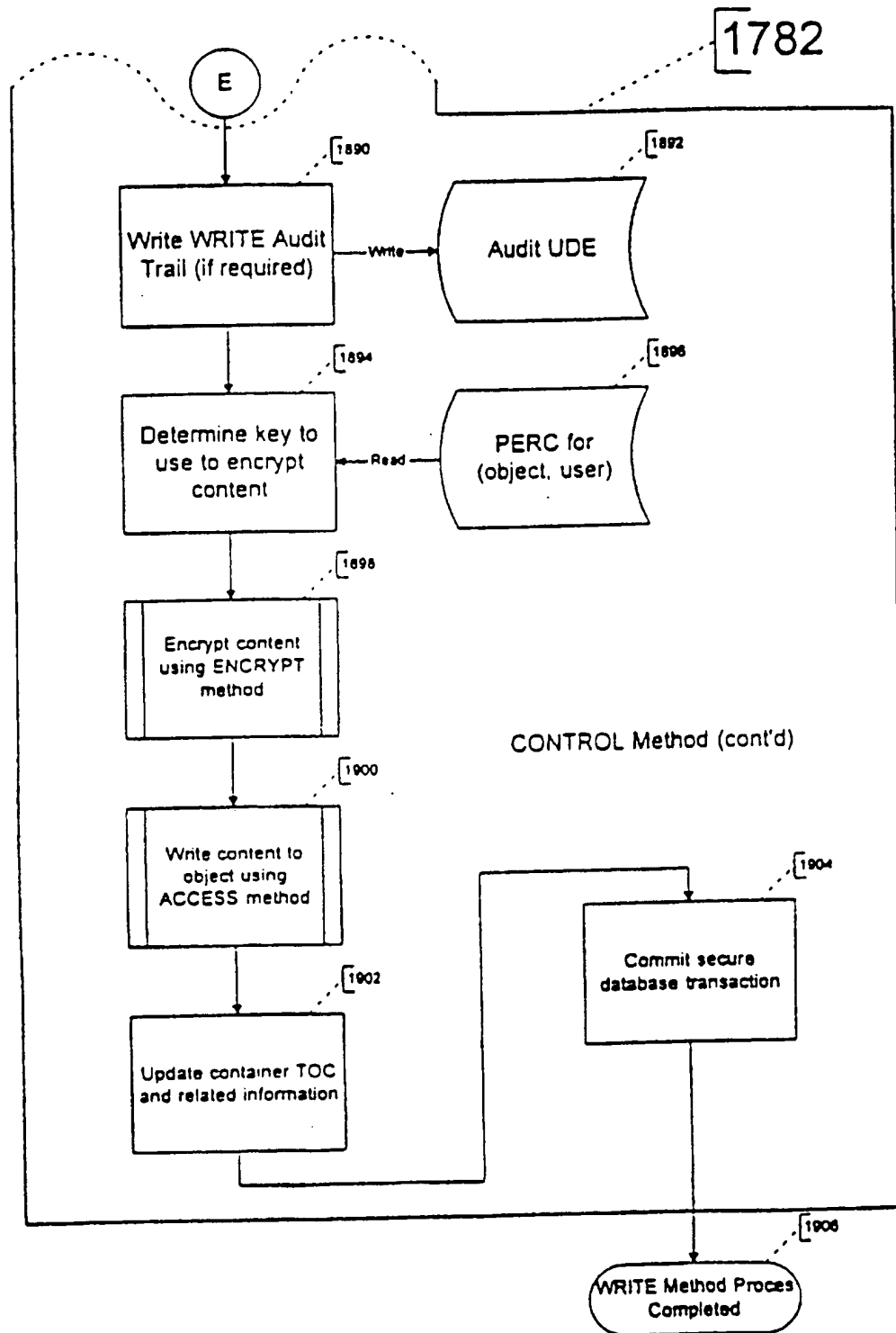
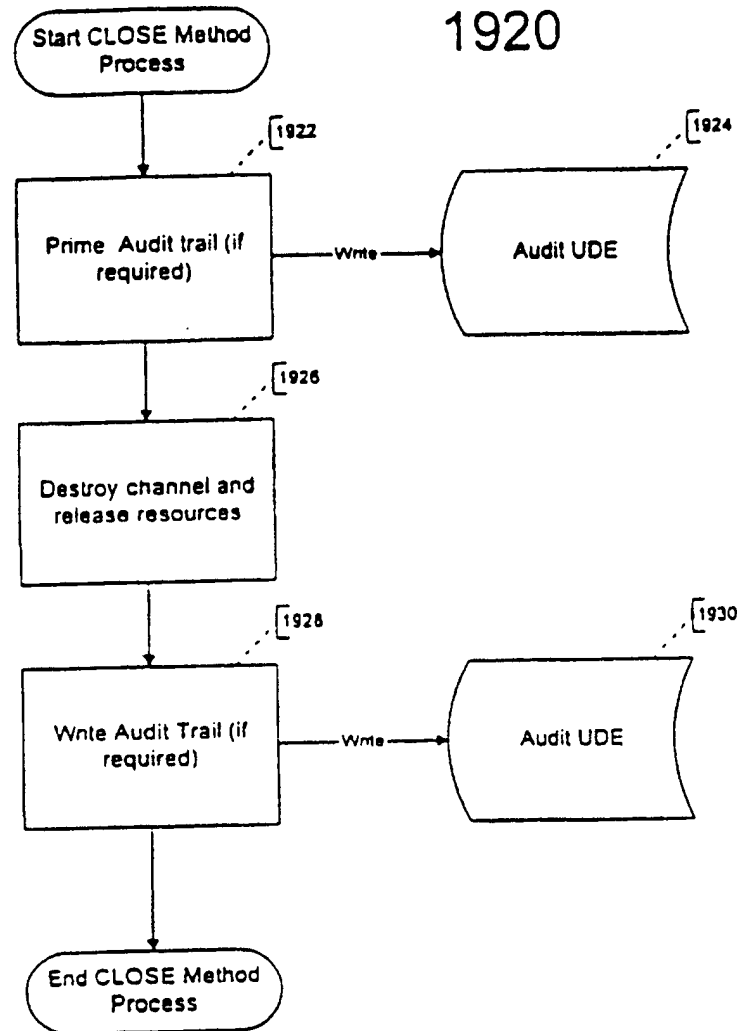


Figure 51f

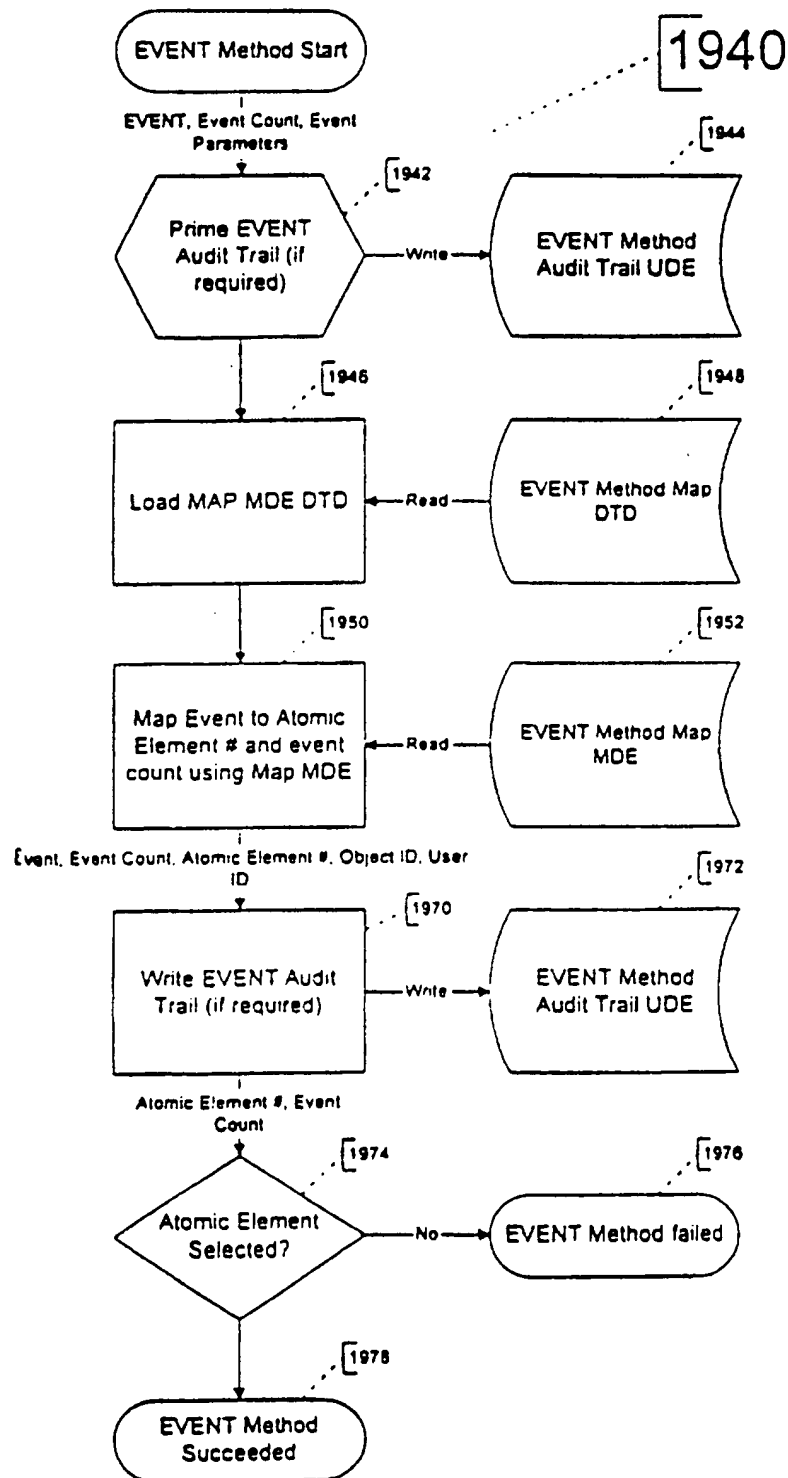
100/163



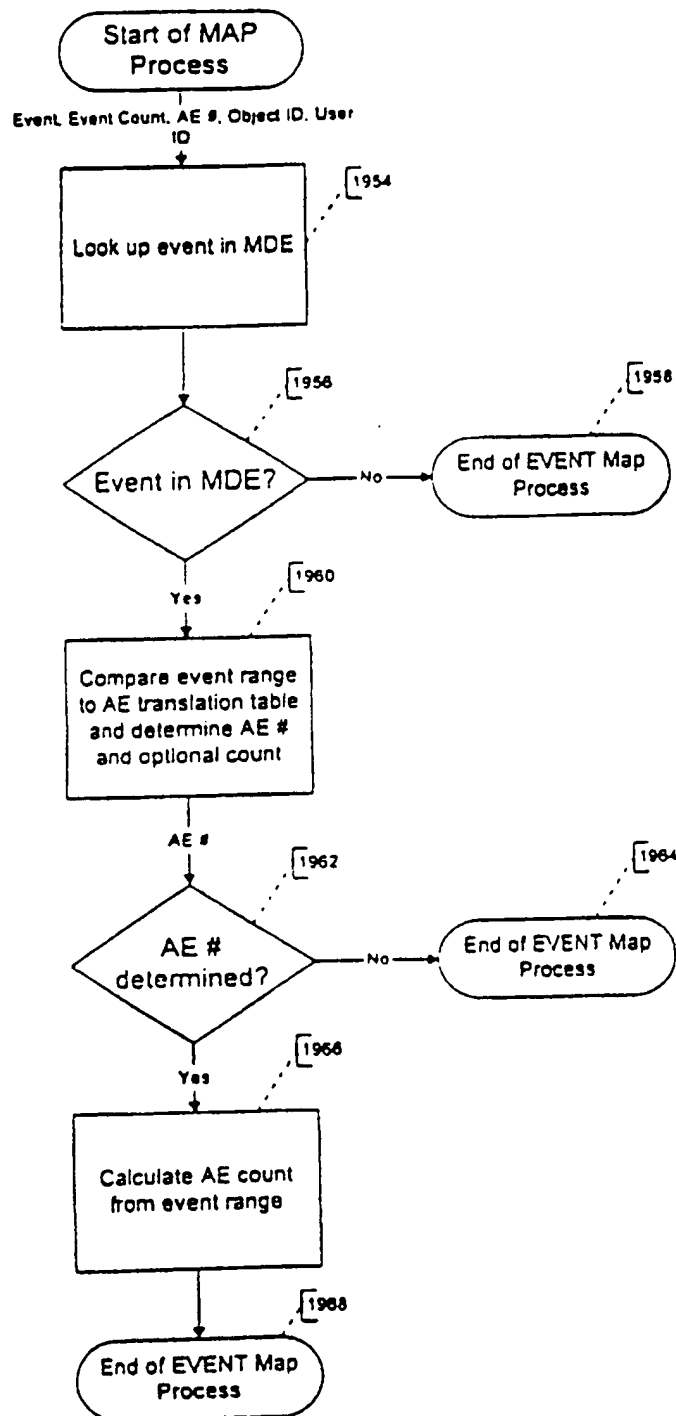
CLOSE  
Method  
Process Flow

Figure 52

101/163

EVENT  
Method  
Process  
Flows

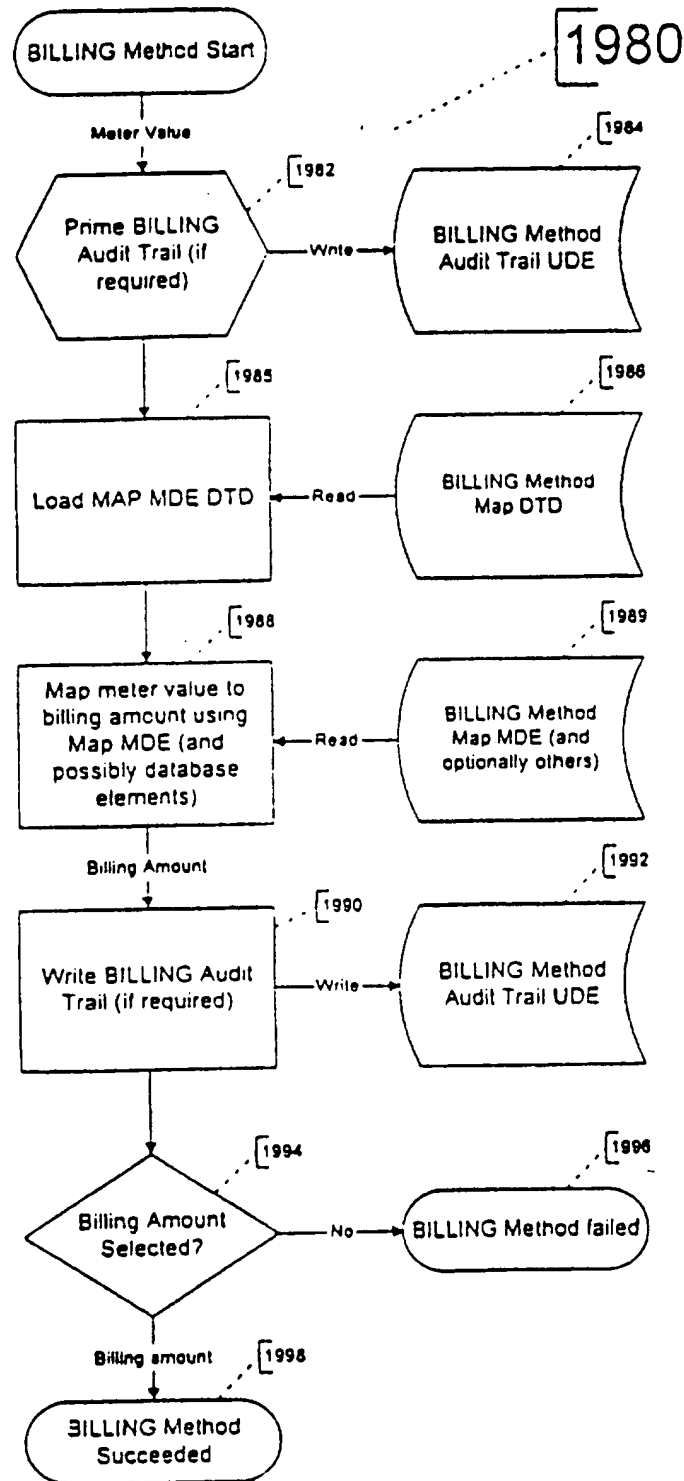
102/163

Sample  
EVENT  
Method  
Mapping  
Process



103/163

# BILLING Method Process Flows



104/163

# ACCESS Method Process Flow

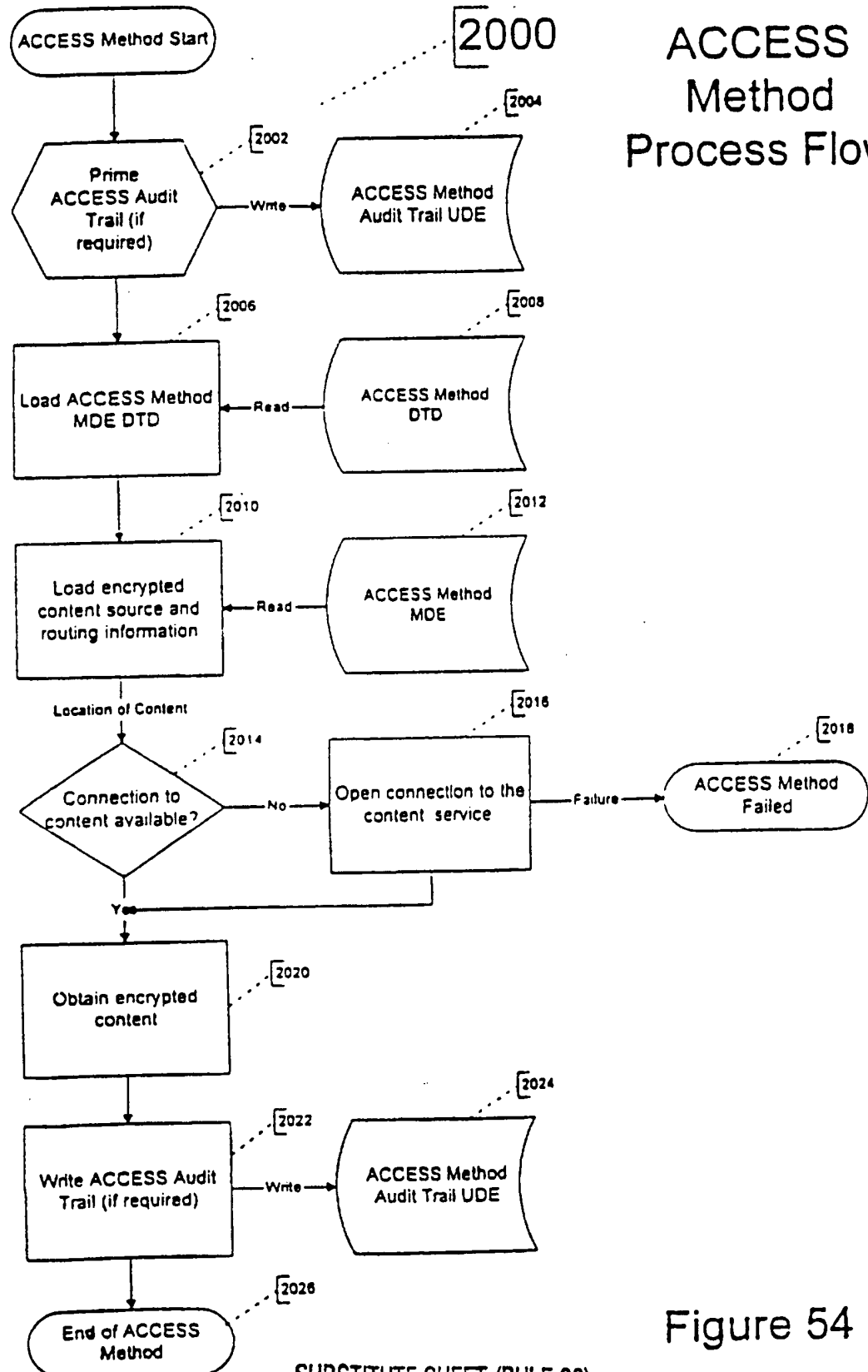
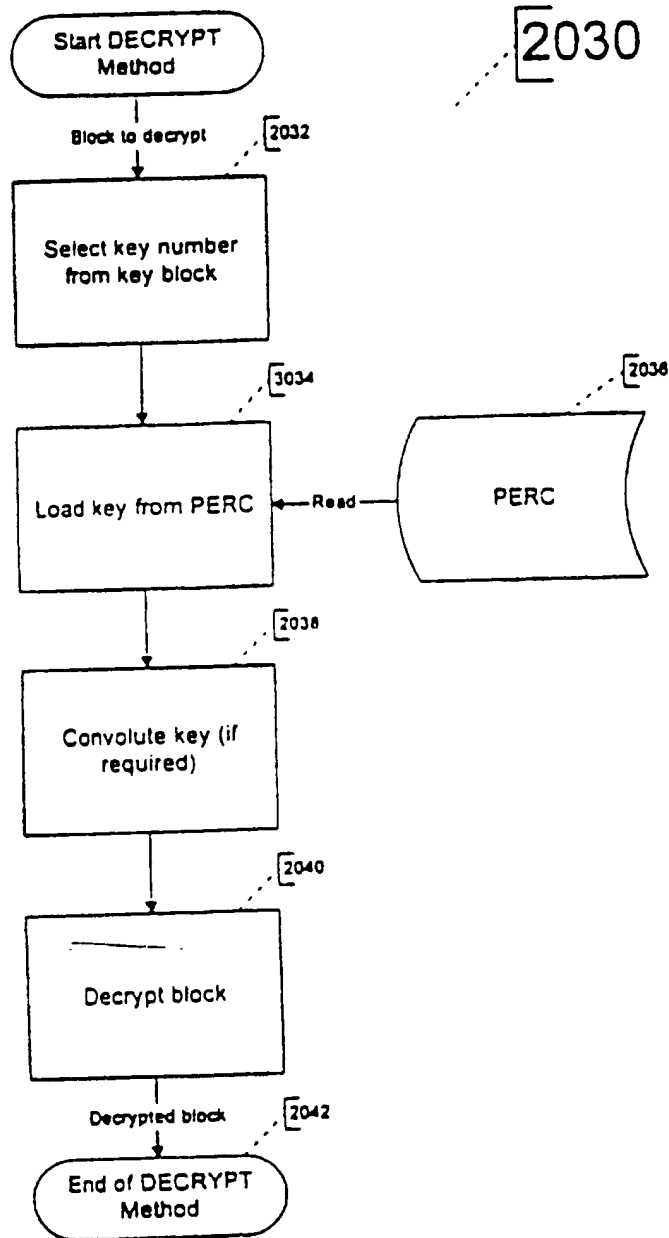


Figure 54

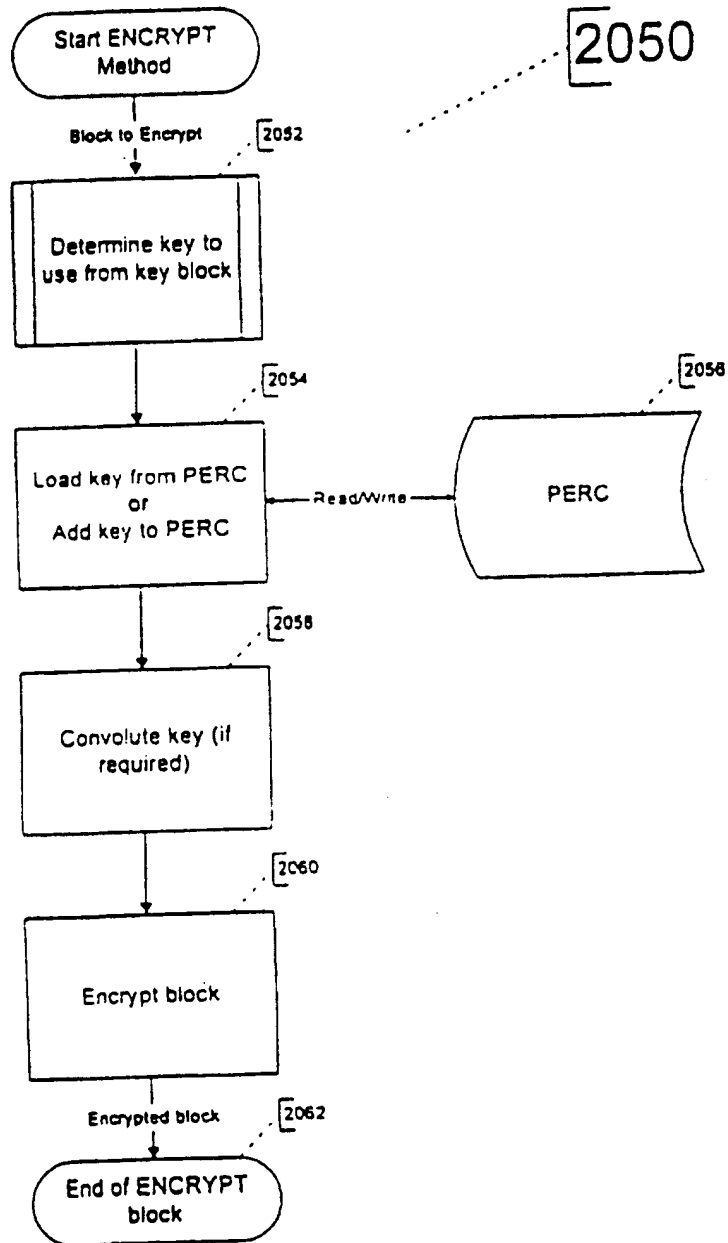
105/163

# DECRYPT Method Process Flow



106/163

# ENCRYPT Method Process Flow



107/163

# CONTENT Method Process Flow

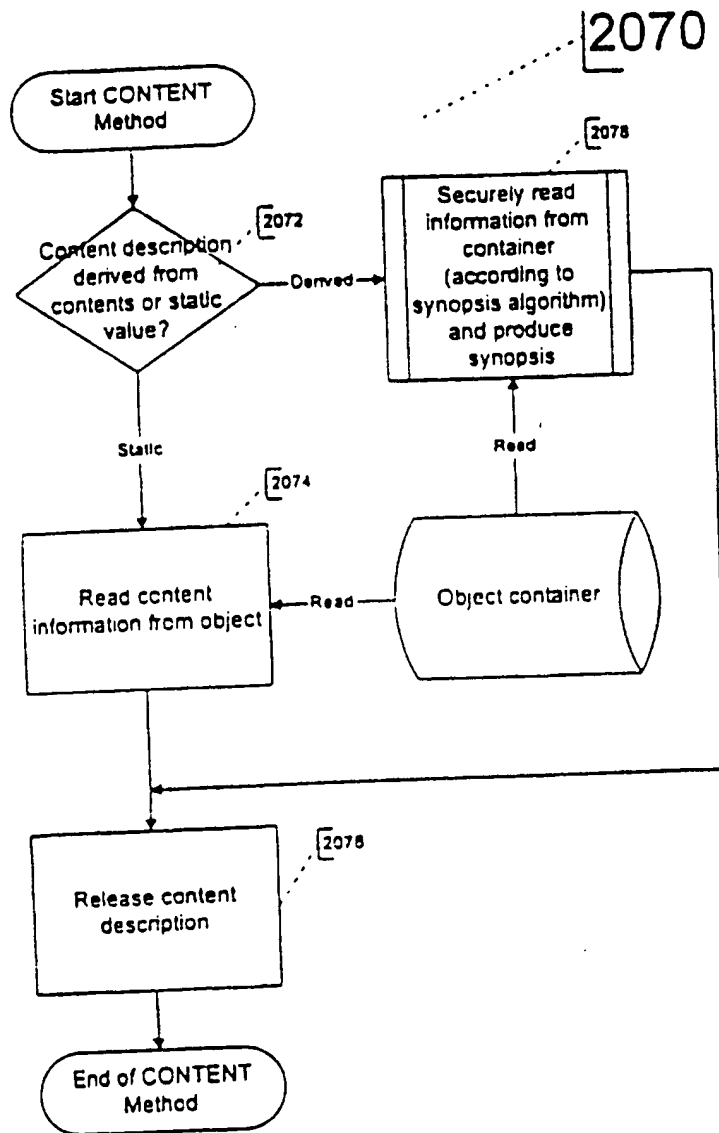
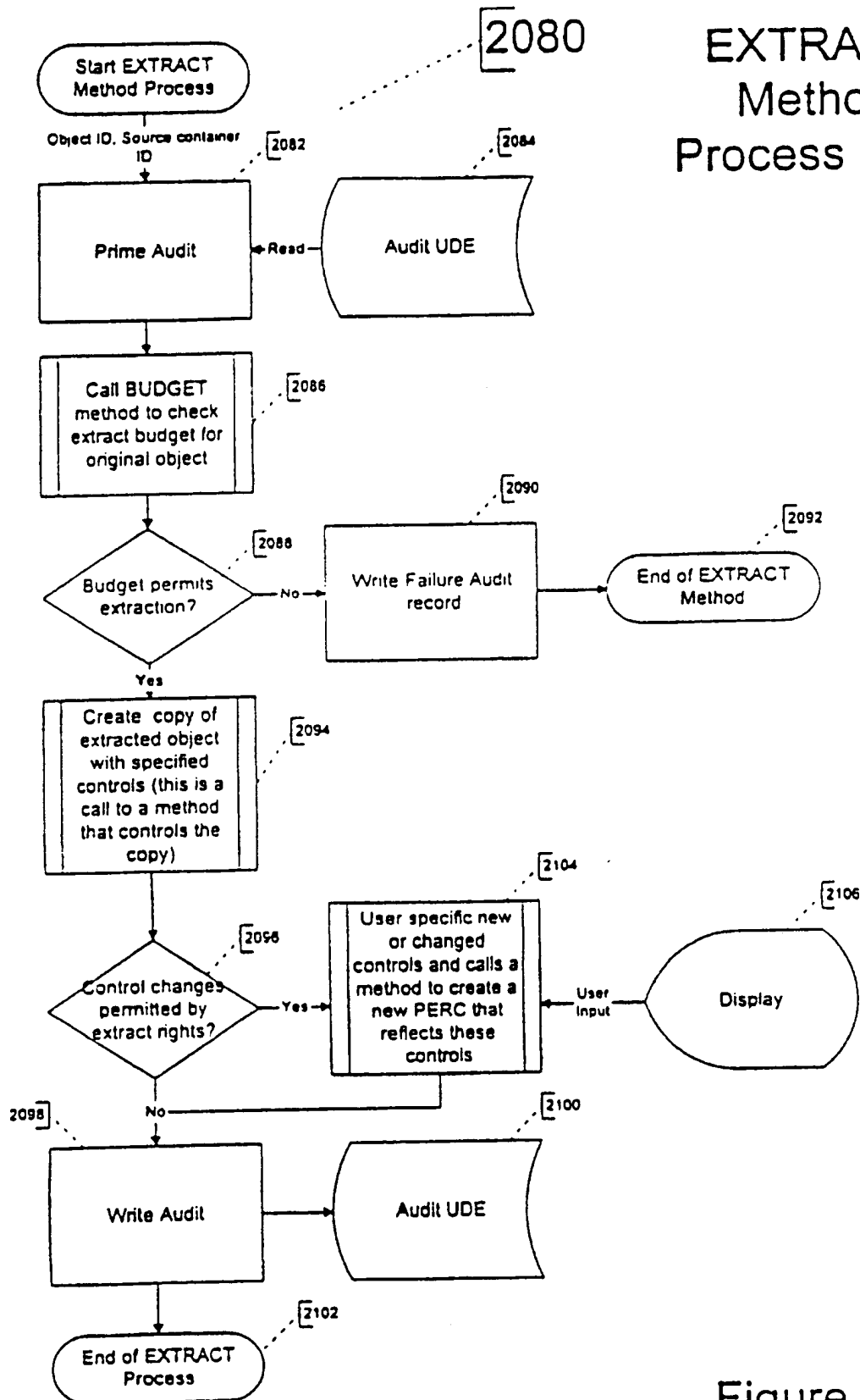


Figure 56

108/163

# EXTRACT Method Process Flow



109/163

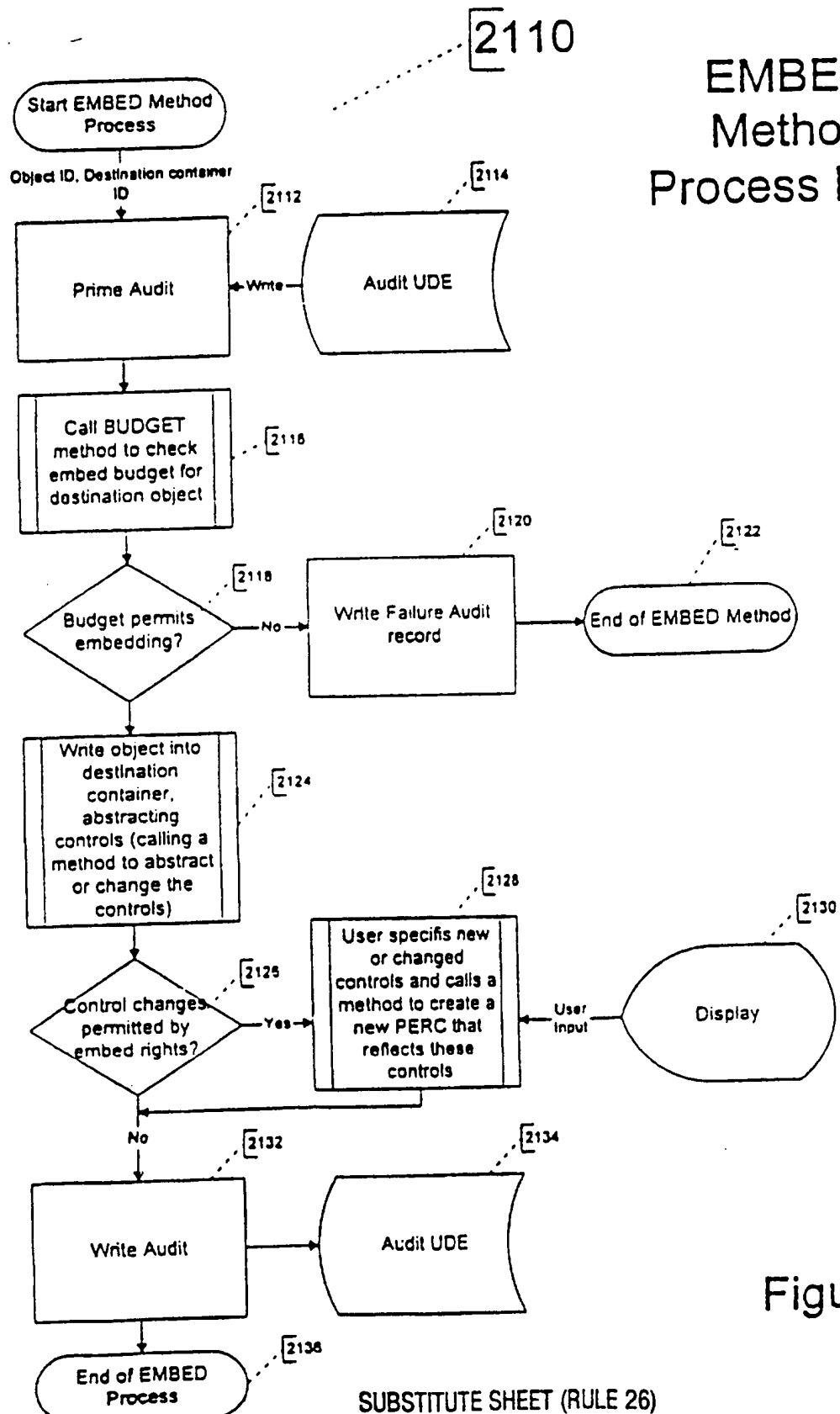
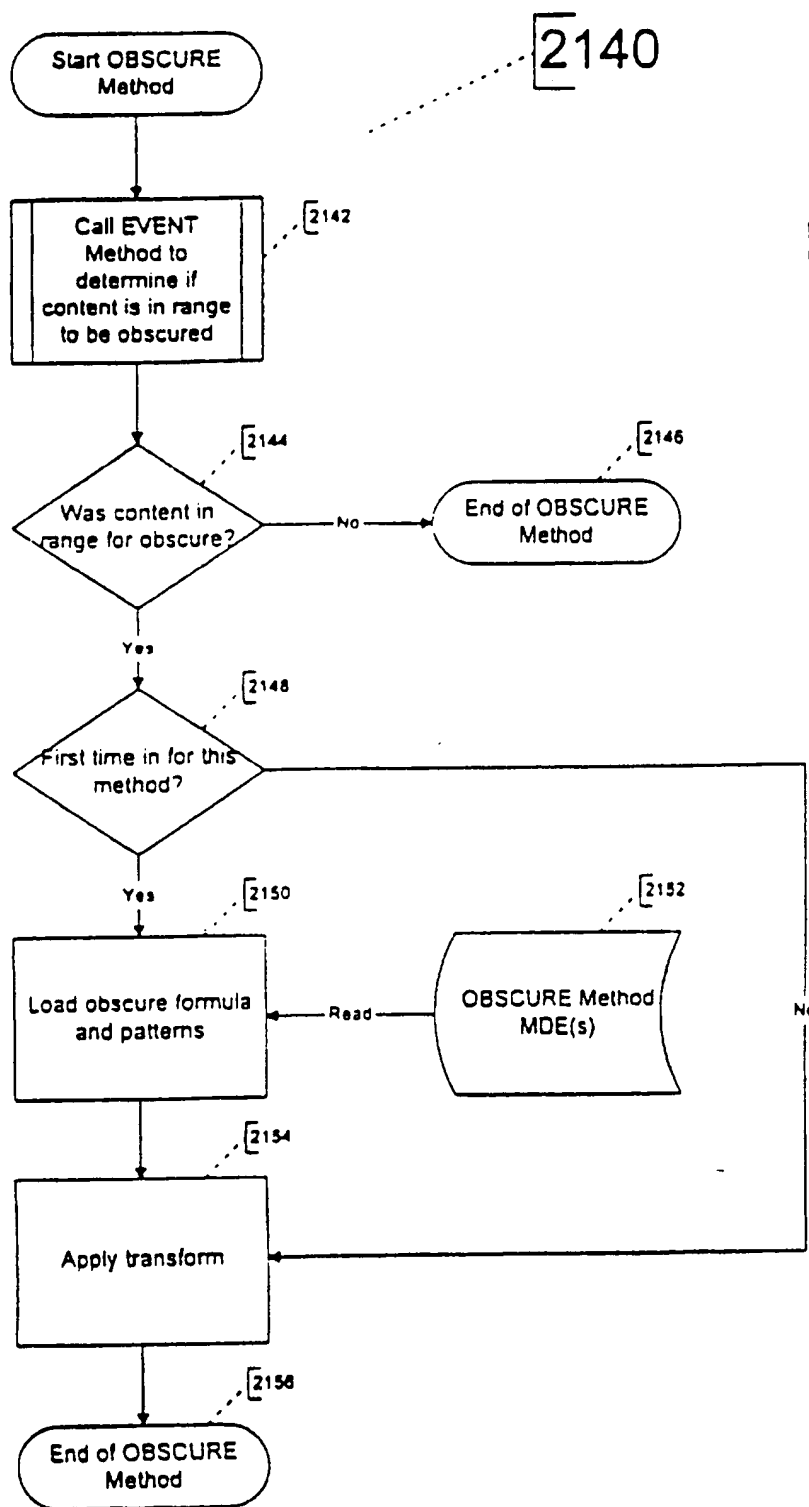


Figure 57b

110/163



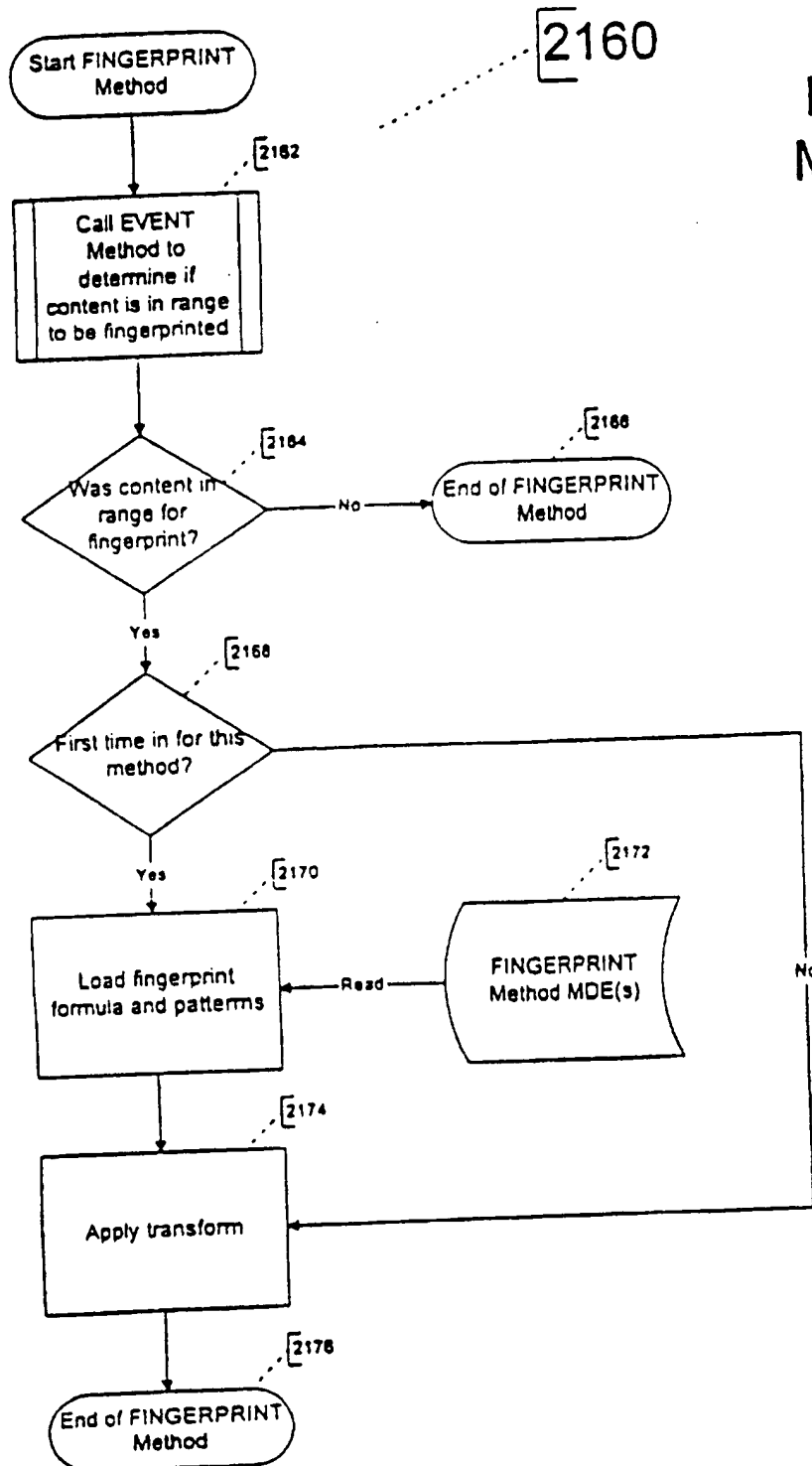
## OBSCURE Method Process Flow

Figure 58a



111/163

# FINGERPRINT Method Process Flow



112/163

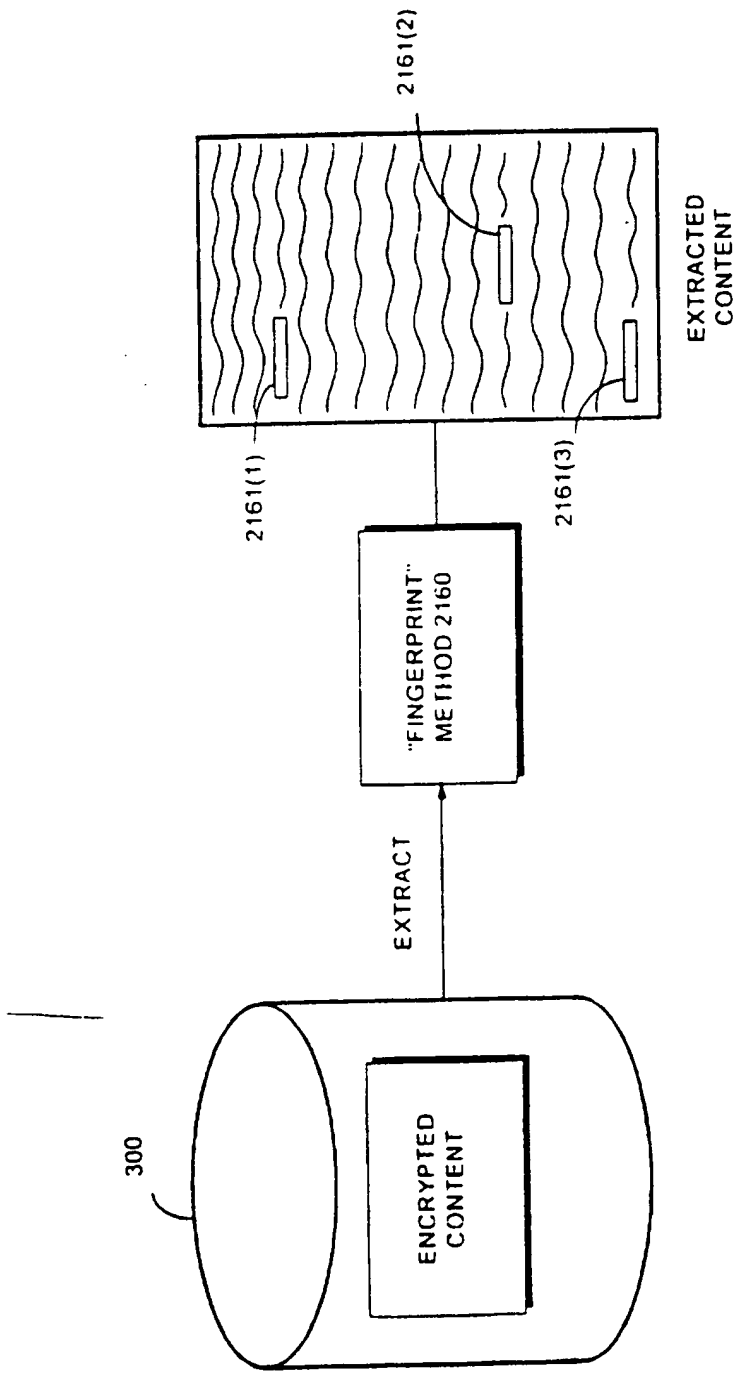


FIG. 58C

113/163

# DESTROY Method Process Flow

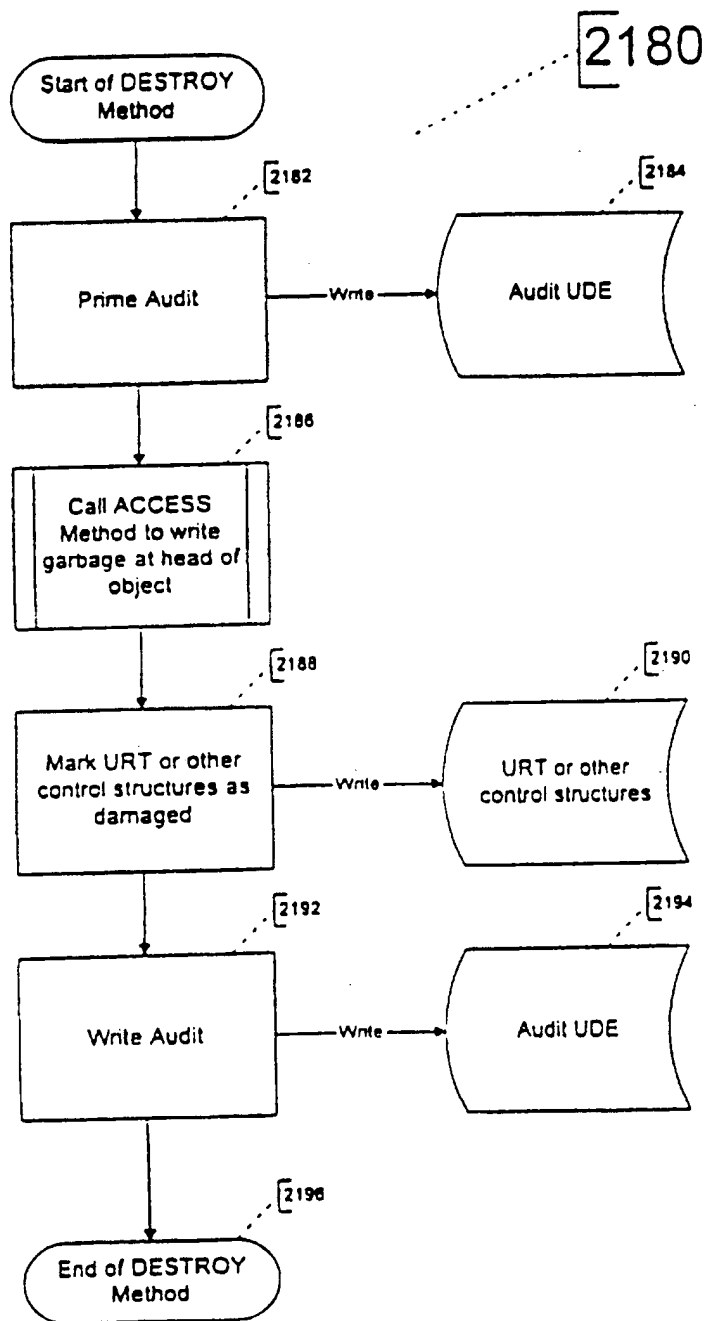
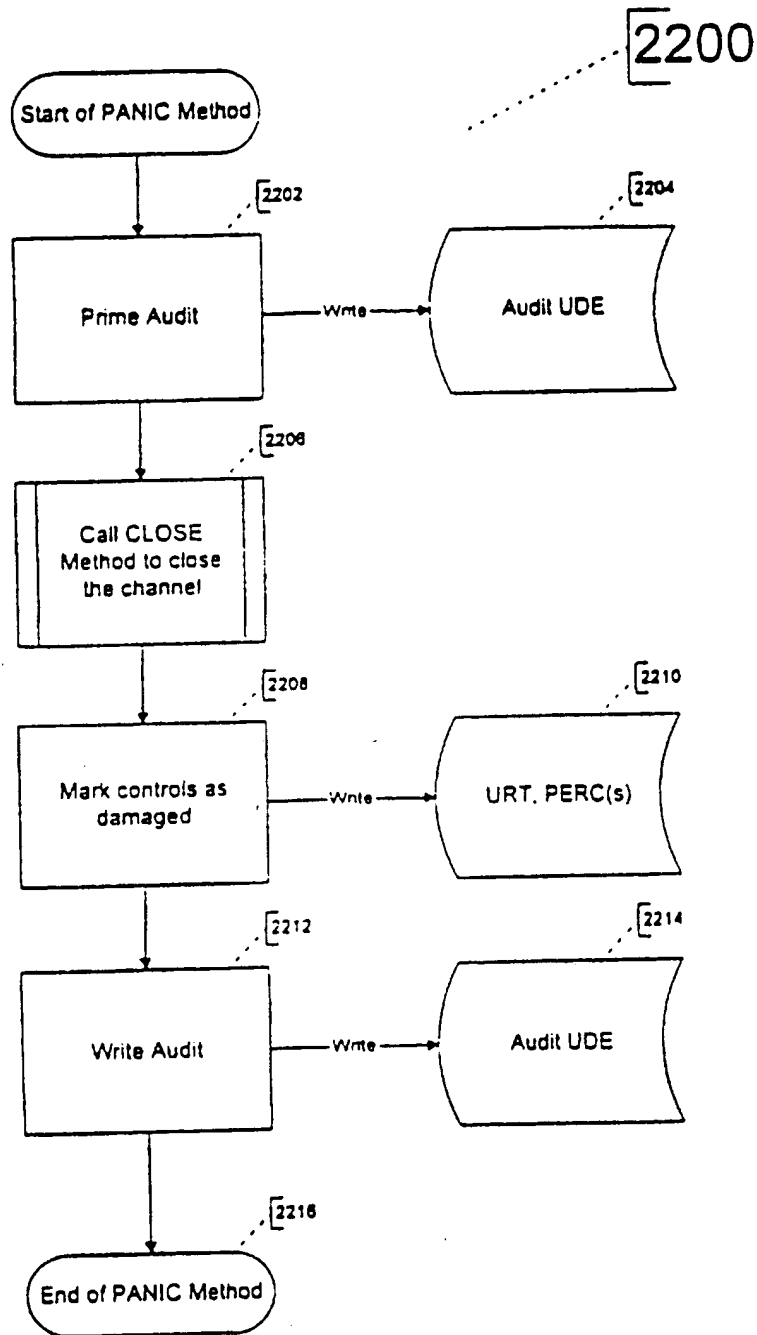


Figure 59

114/163



## PANIC Method Process Flow

METER  
Method Use  
Process Flow

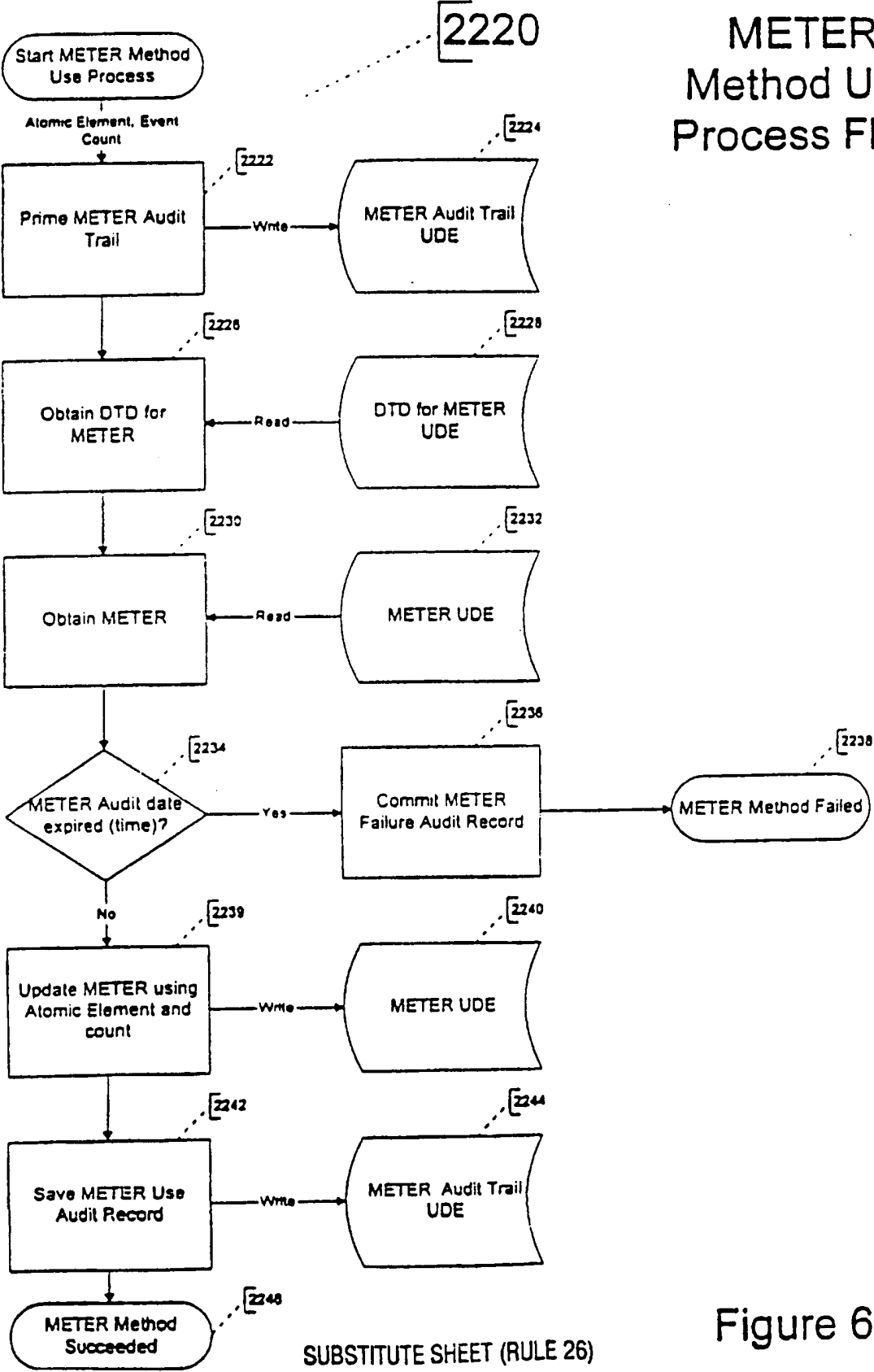
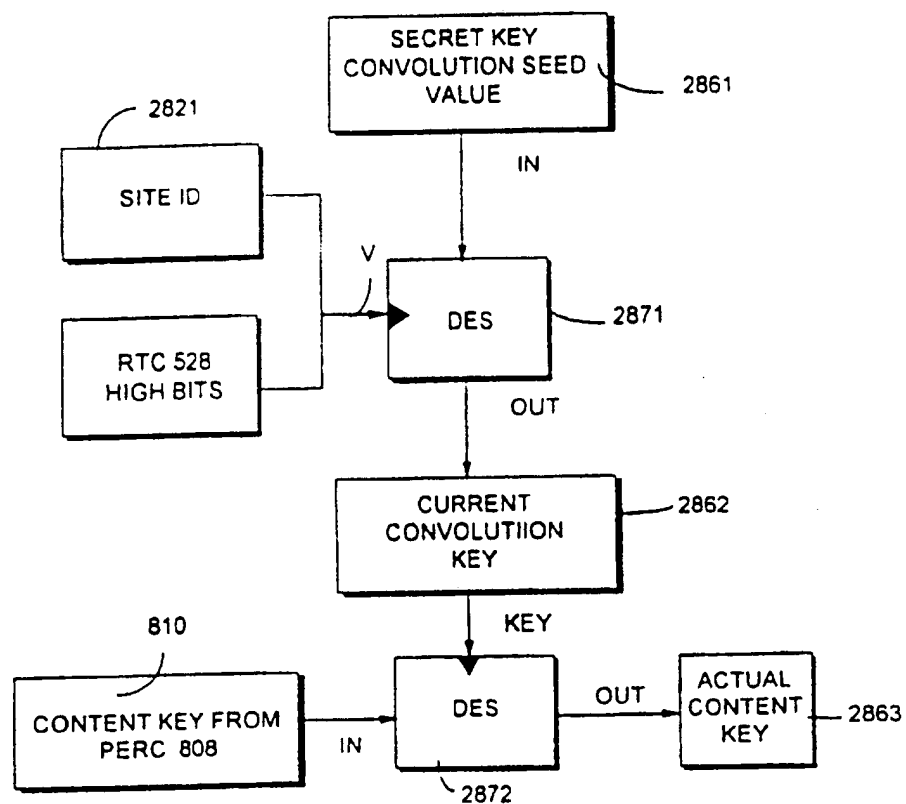


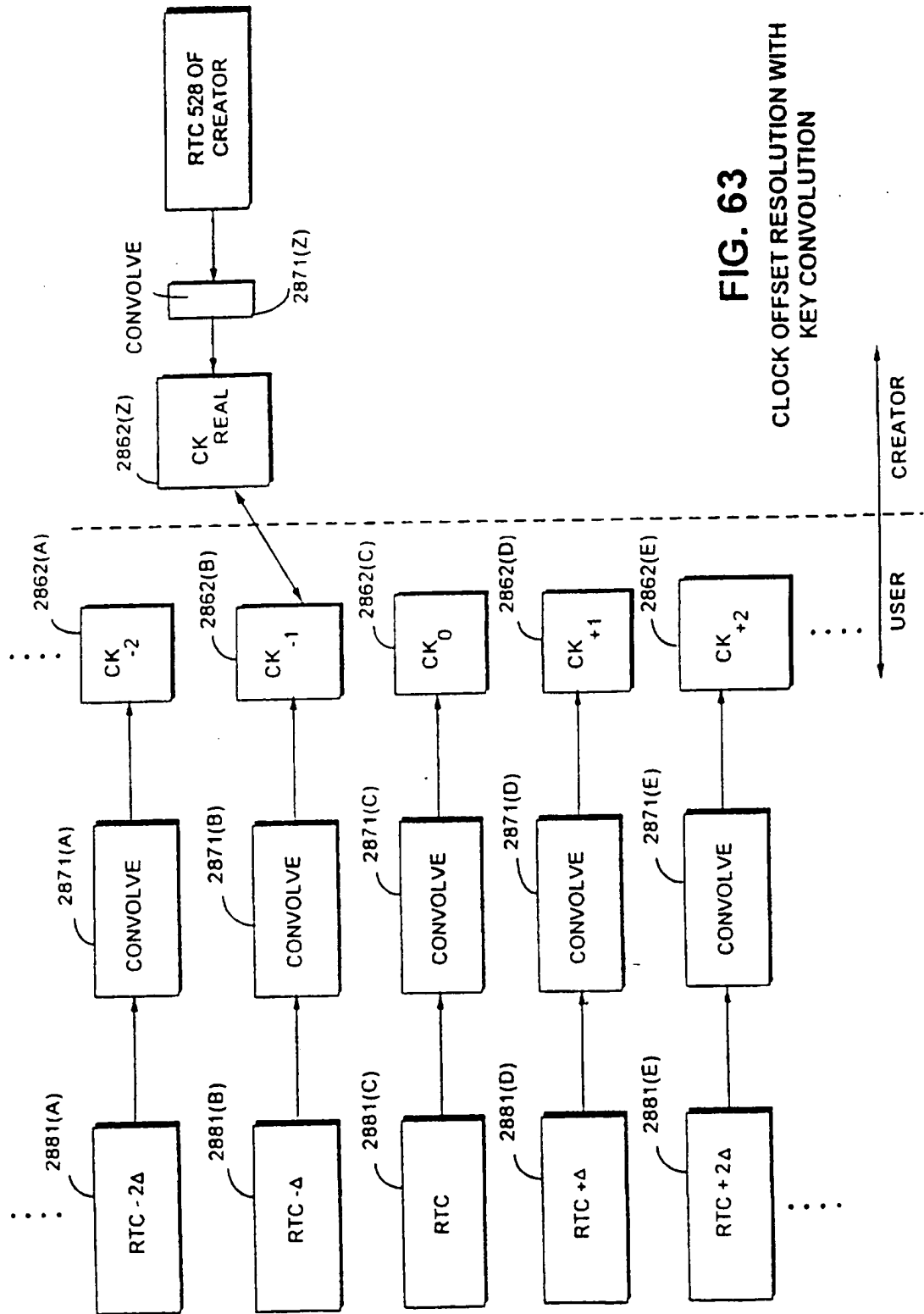
Figure 61

116/163

**FIG. 62**  
KEY CONVOLUTION PROCESS

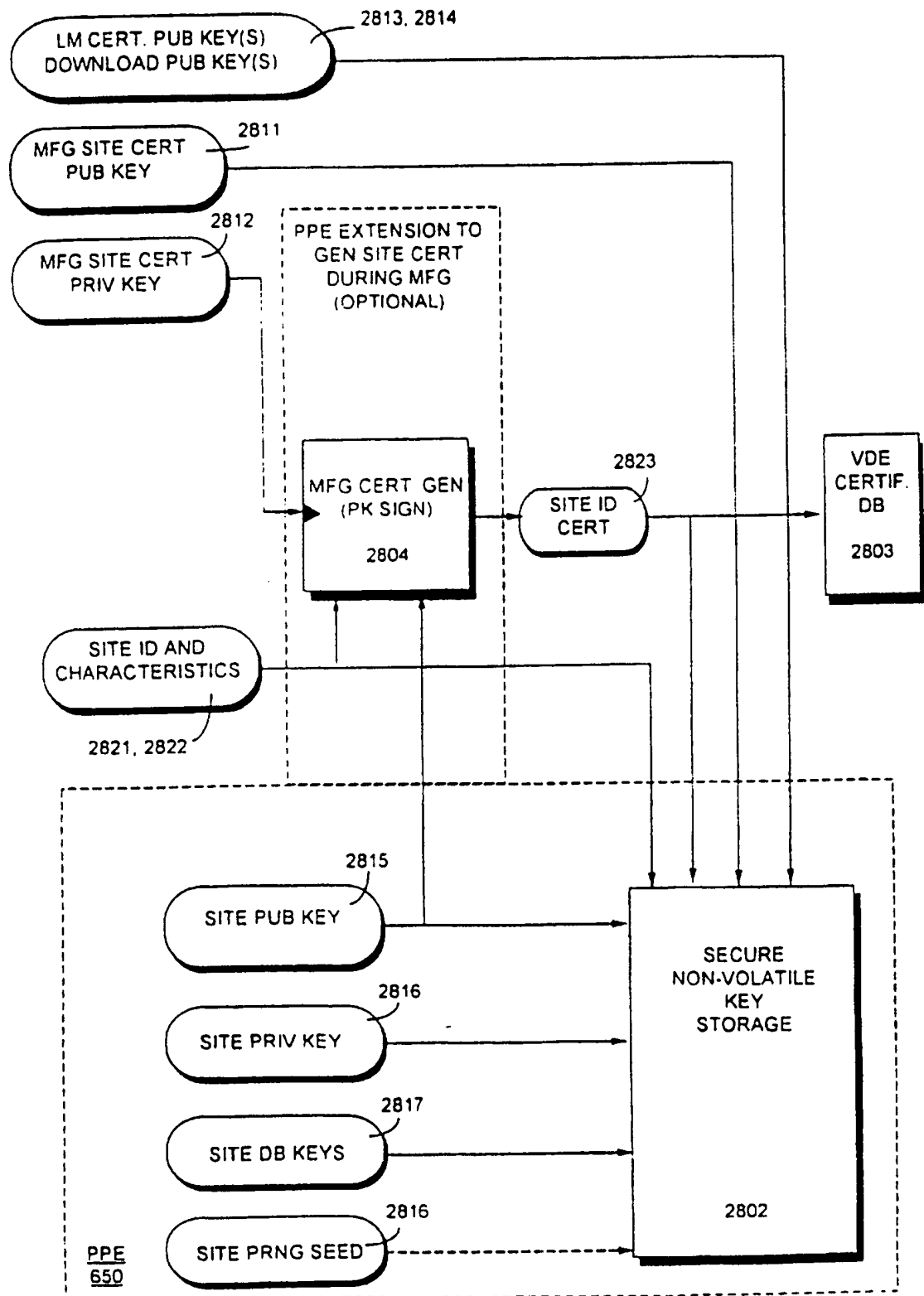


117/163



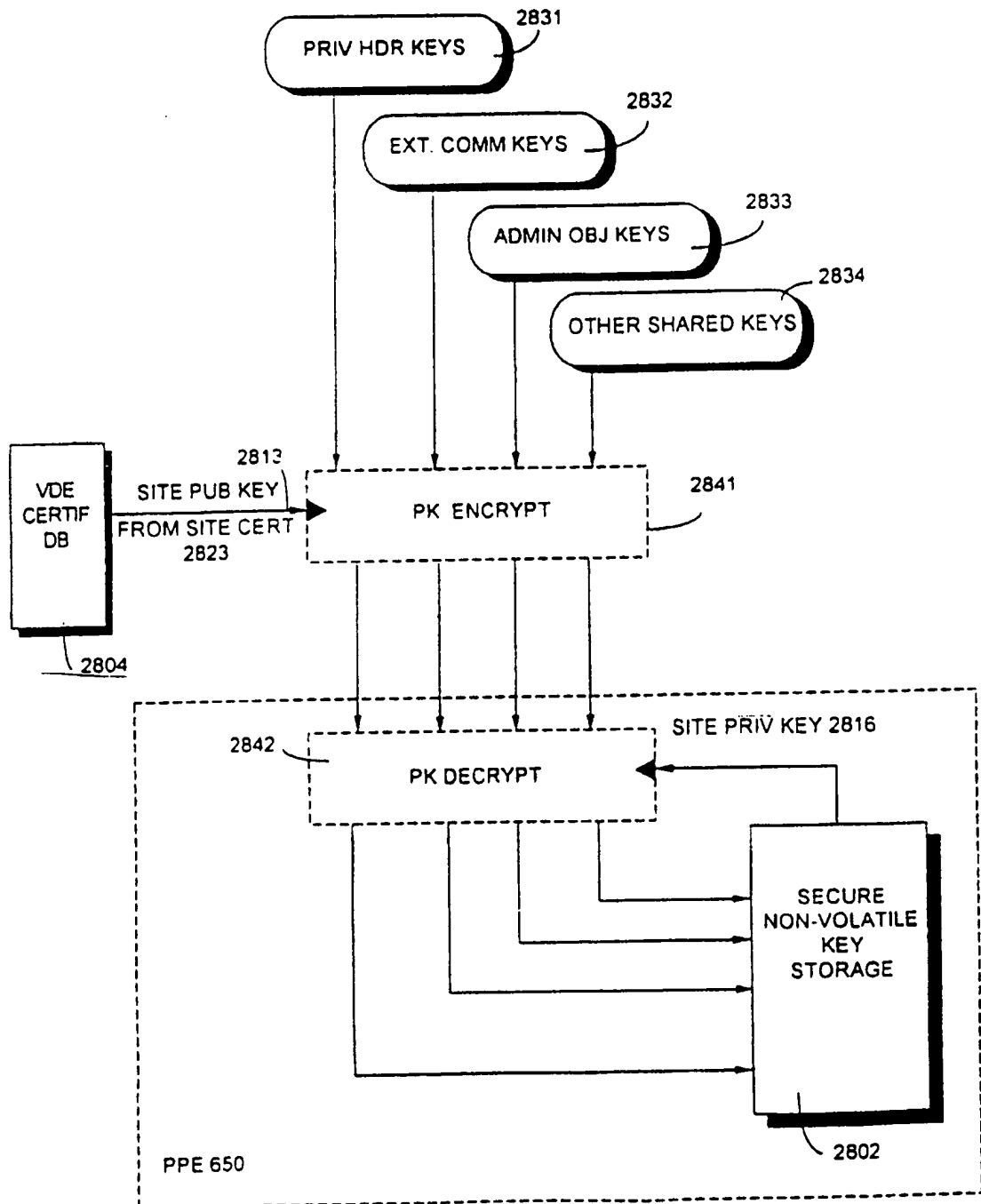
**FIG. 63**  
CLOCK OFFSET RESOLUTION WITH  
KEY CONVOLUTION

118/163

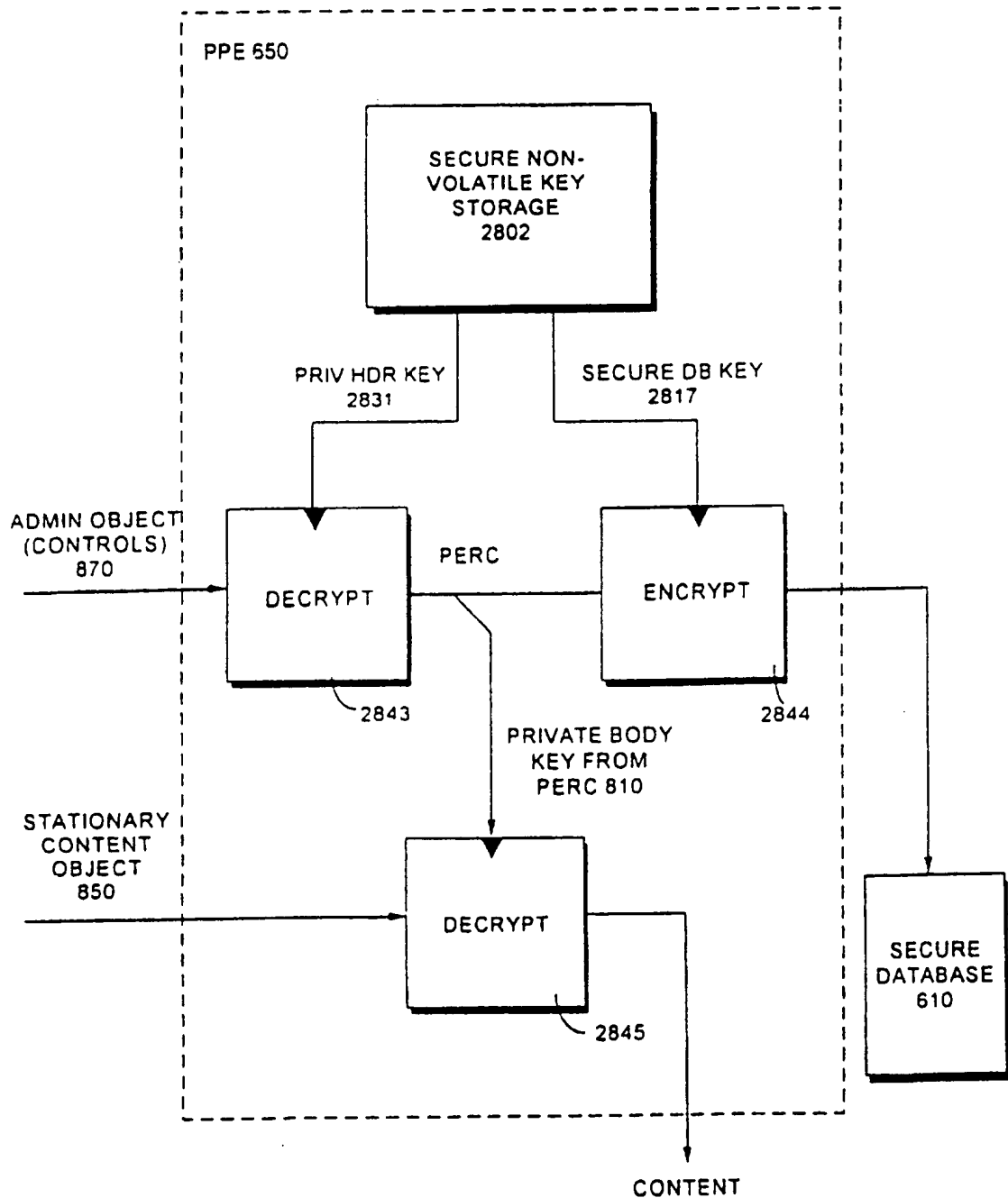
**FIG. 64** SPU KEY INITIALIZATION/INSTALLATION



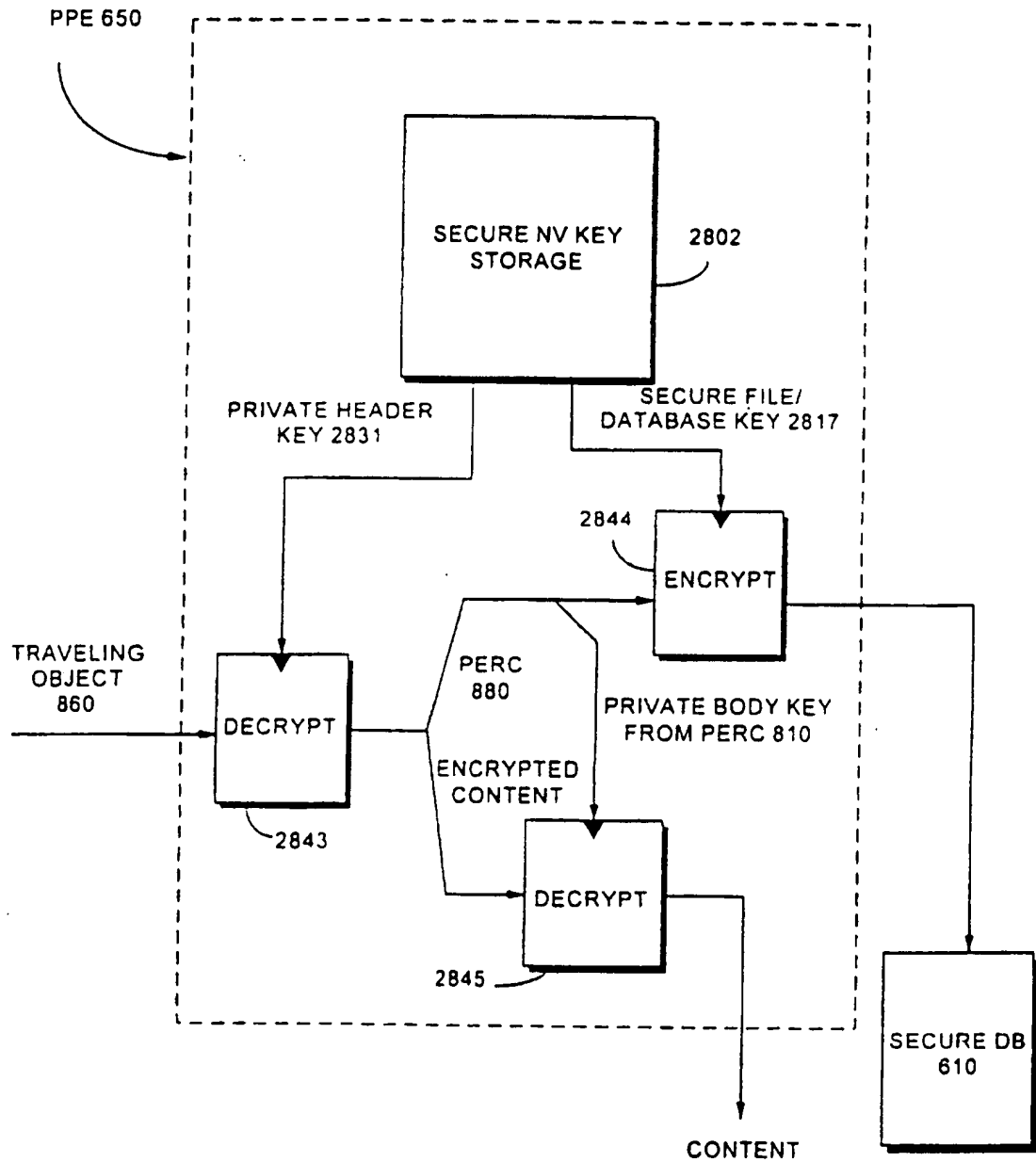
119/163

**FIG. 65** KEY INSTALLATION & UPDATE

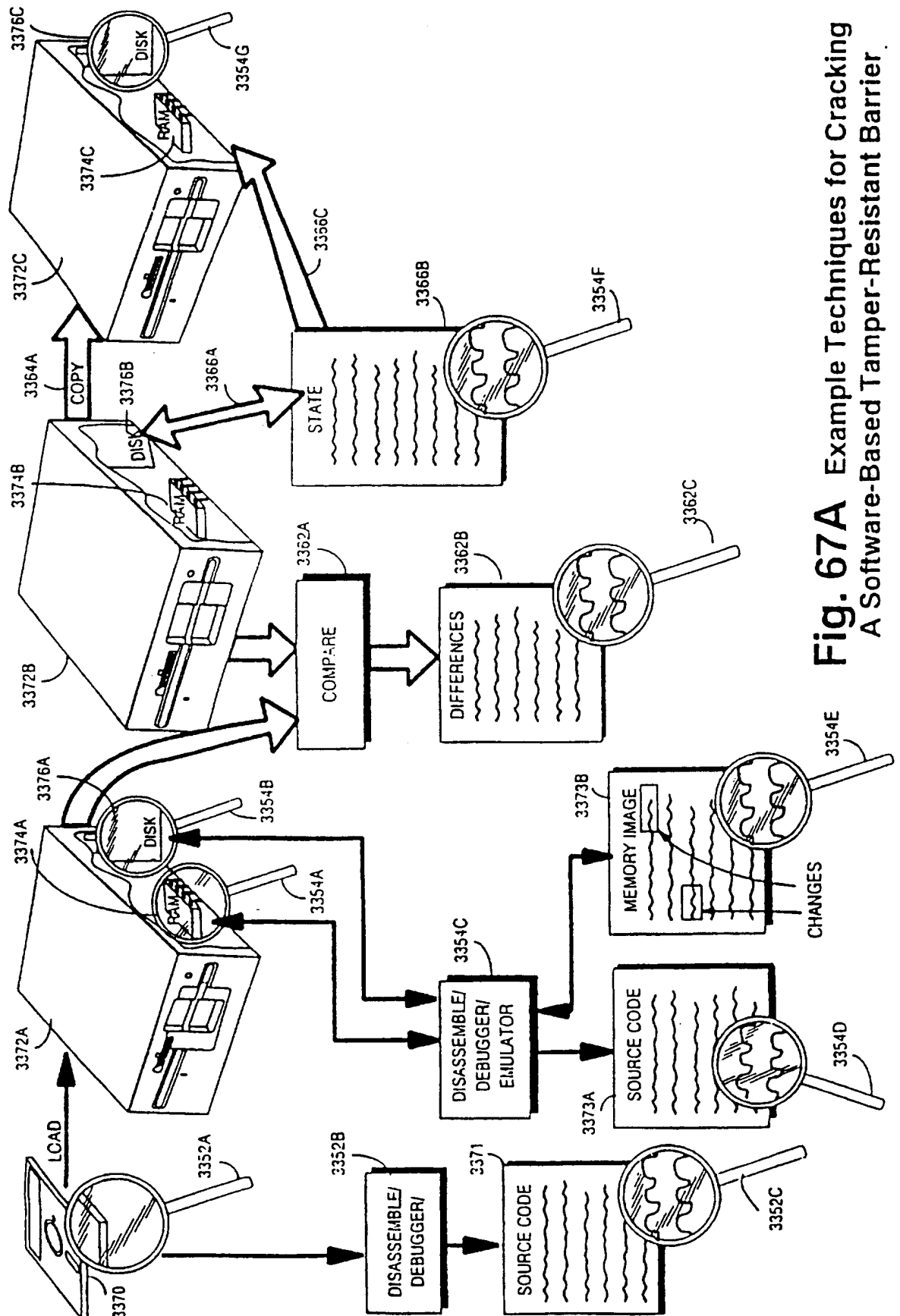
120/163

**FIG. 66** STATIONARY OBJECT DECRYPTION

121/163

**FIG. 67** TRAVELING OBJECT DECRYPTION

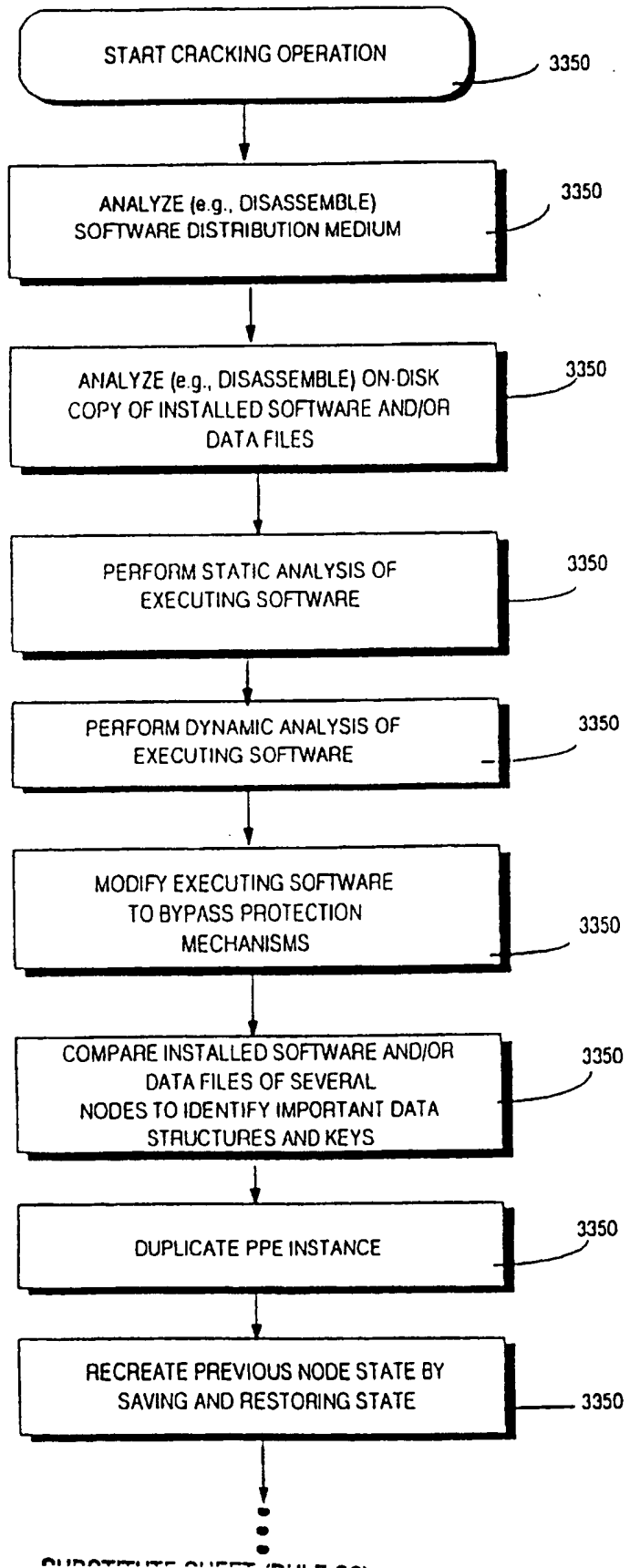
122/163



**Fig. 67A** Example Techniques for Cracking  
A Software-Based Tamper-Resistant Barrier

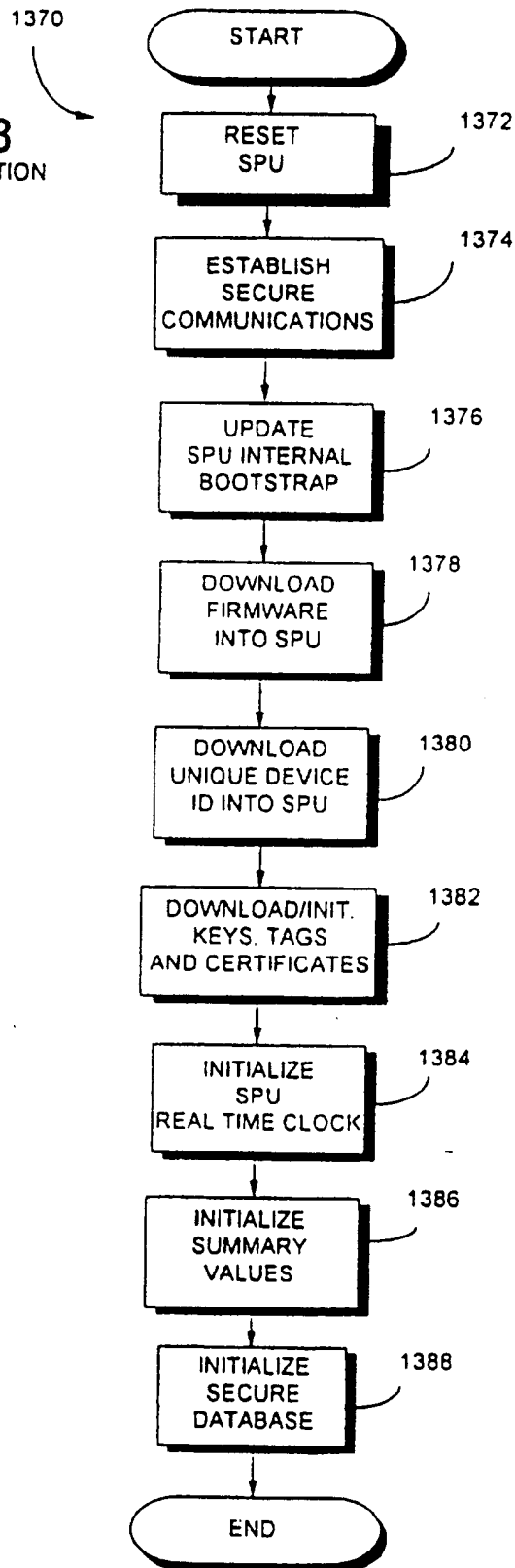
123/163

**Fig. 67B**  
Example Cracking  
Operation

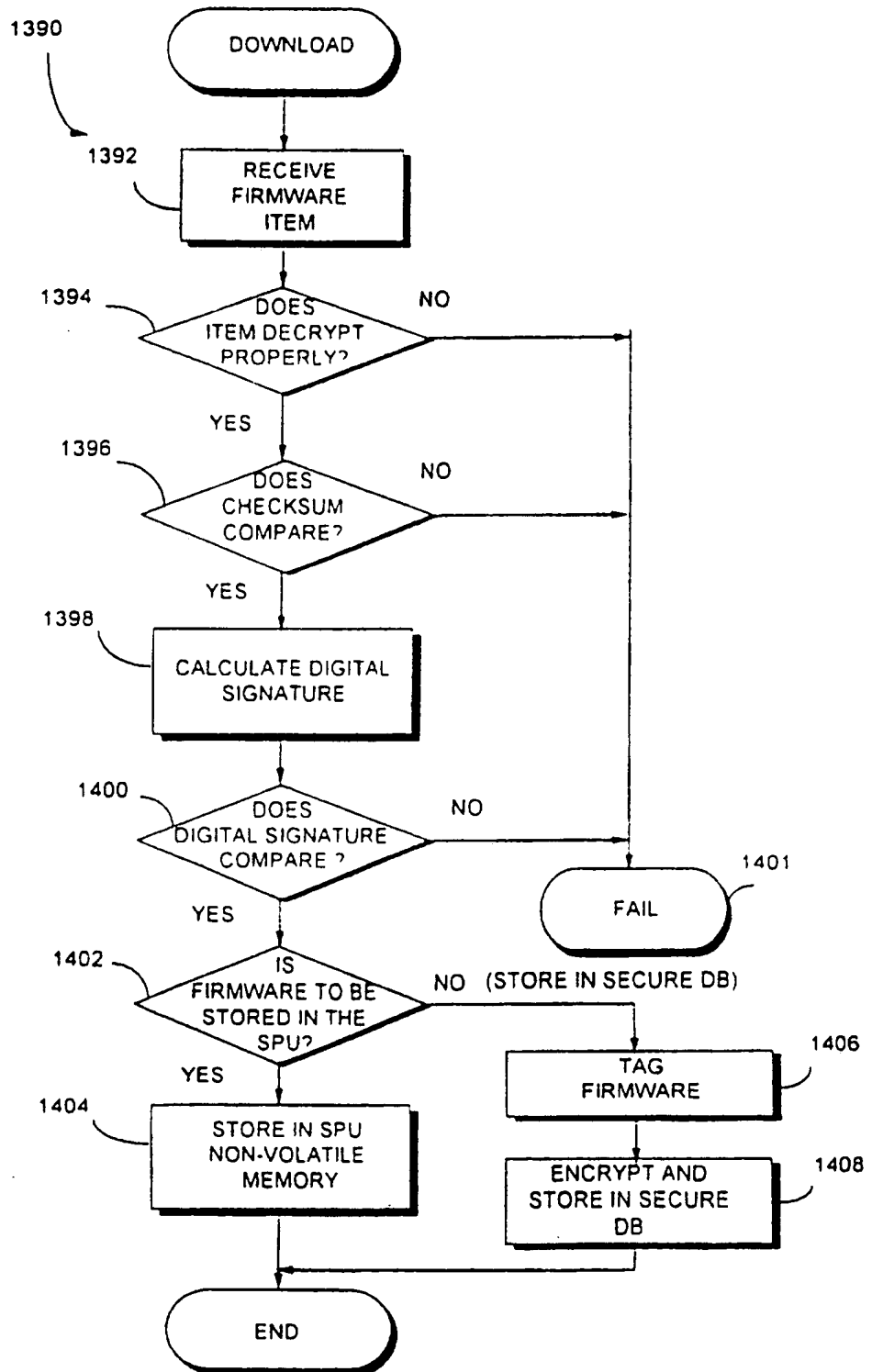


124/163

**FIG. 68**  
SPU INITIALIZATION

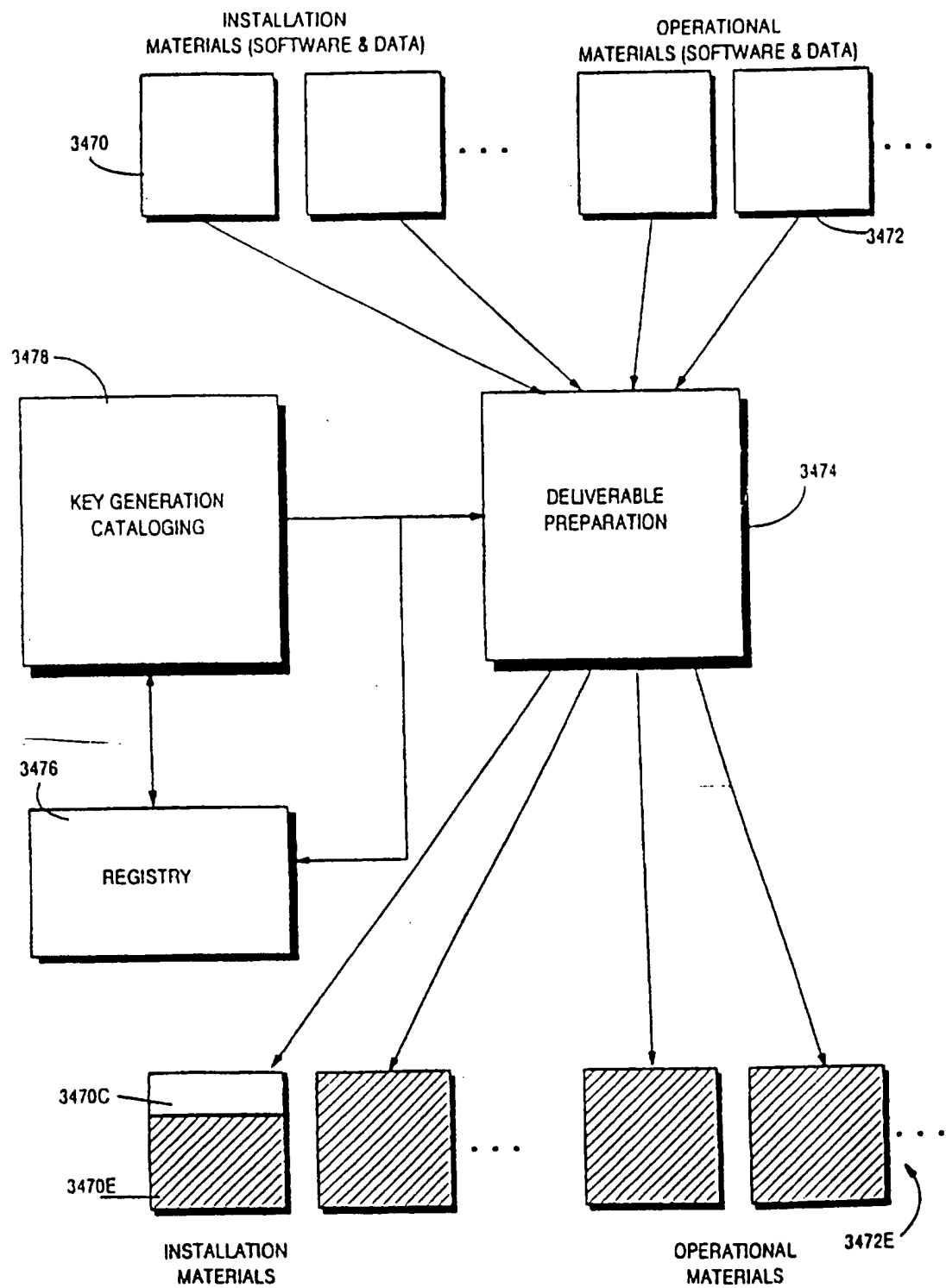


125/163

**FIG. 69**SPU FIRMWARE  
DOWNLOAD

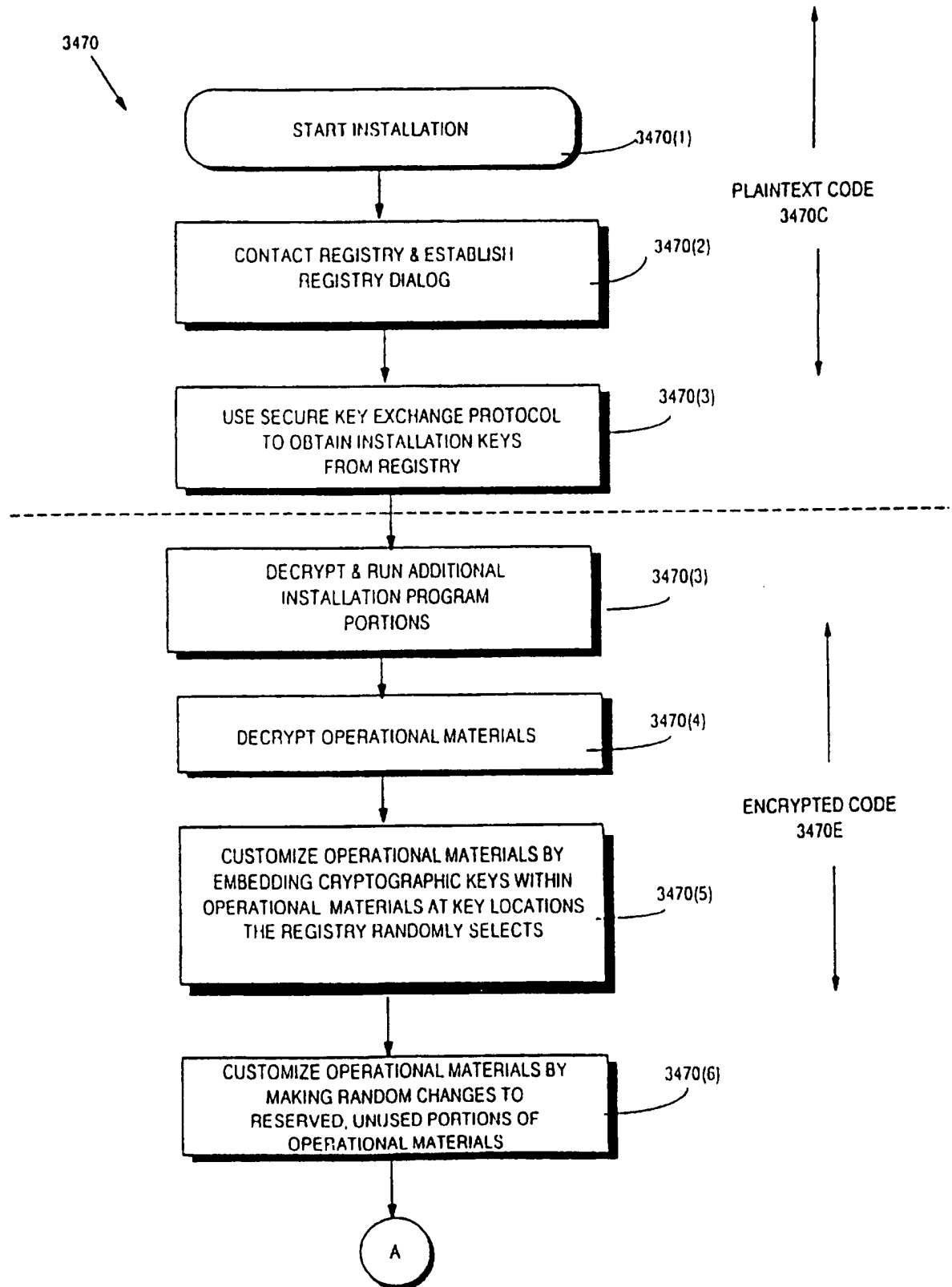
SUBSTITUTE SHEET (RULE 26)

126/163

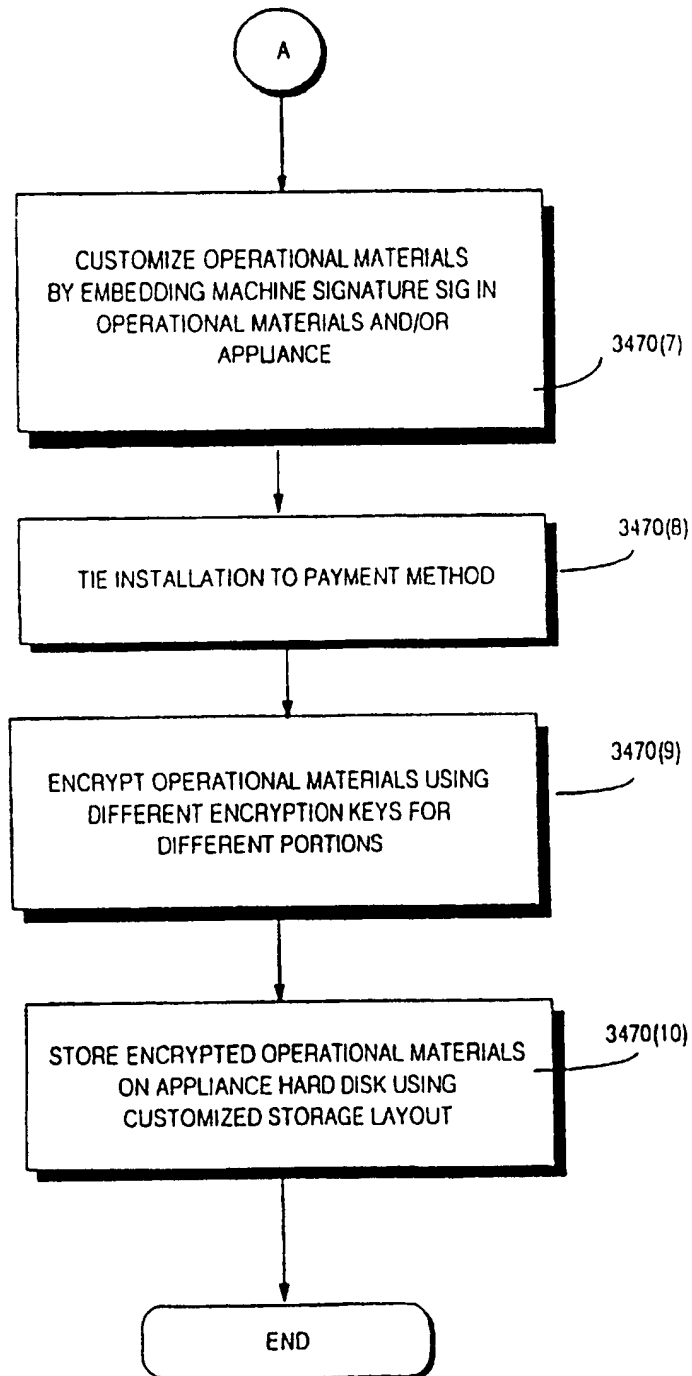
**Fig. 69A** Software Distributed In Encrypted Form



127/163

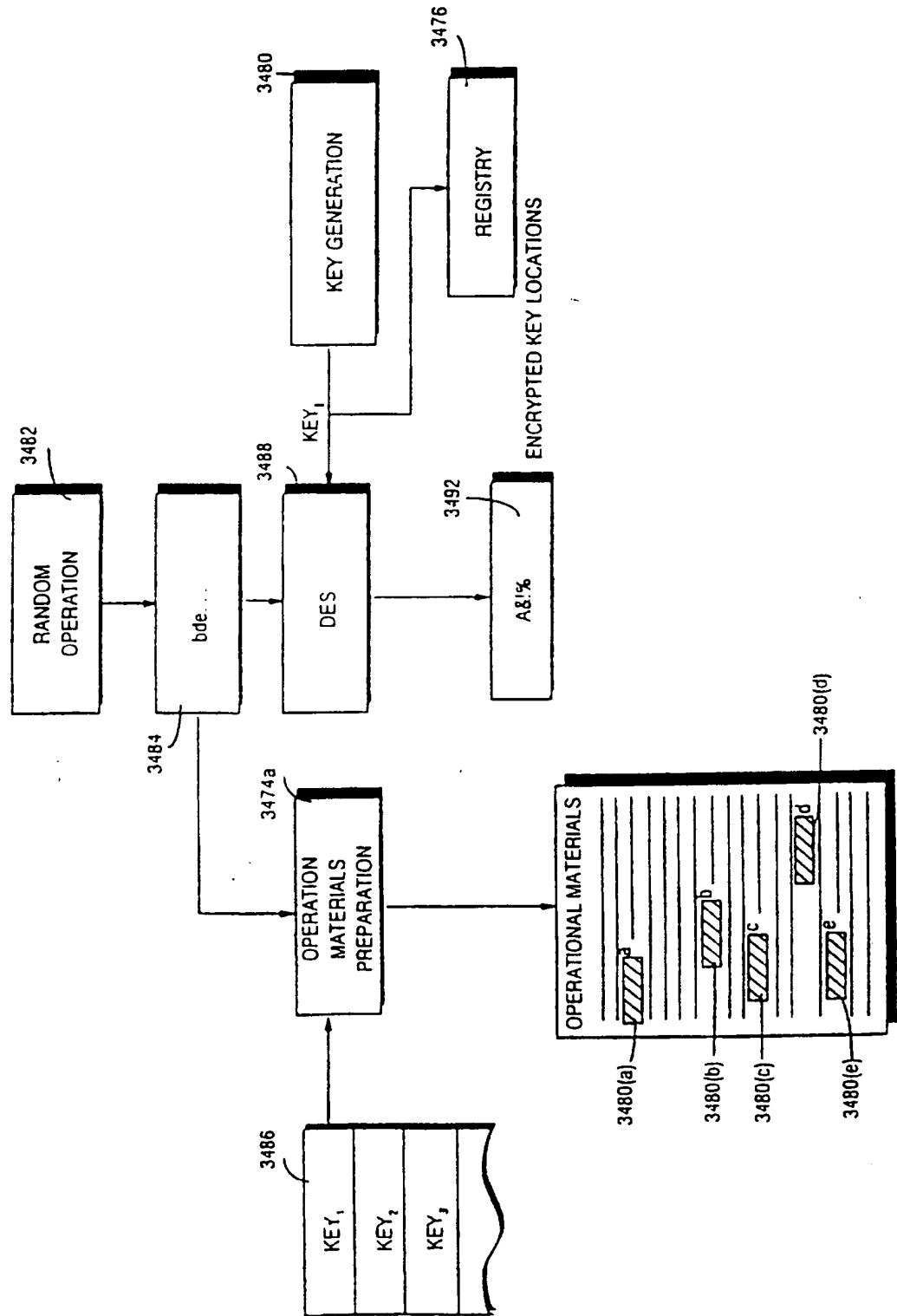
**Fig. 69B** Example Installation Routine

128/163

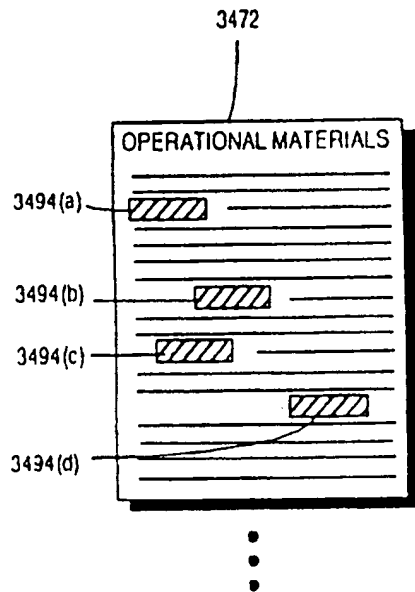
**Fig. 69C** Example Installation Routine

129/163

**Fig. 69D** Embedding Keys At Different Locations With Operational Materials

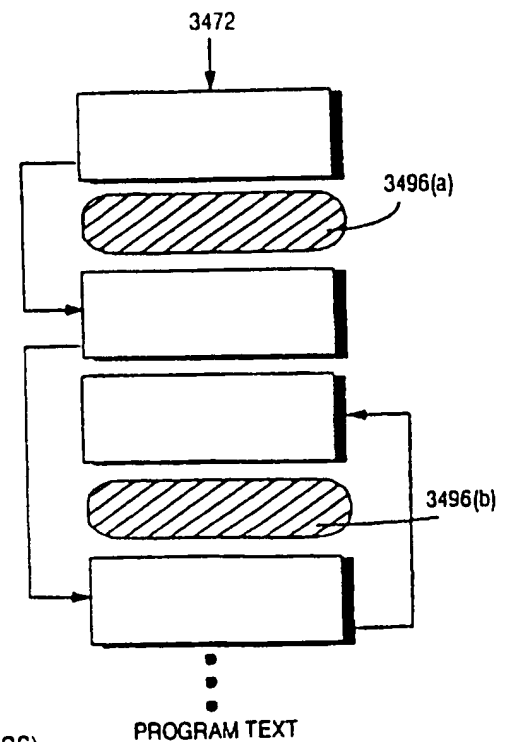


130/163



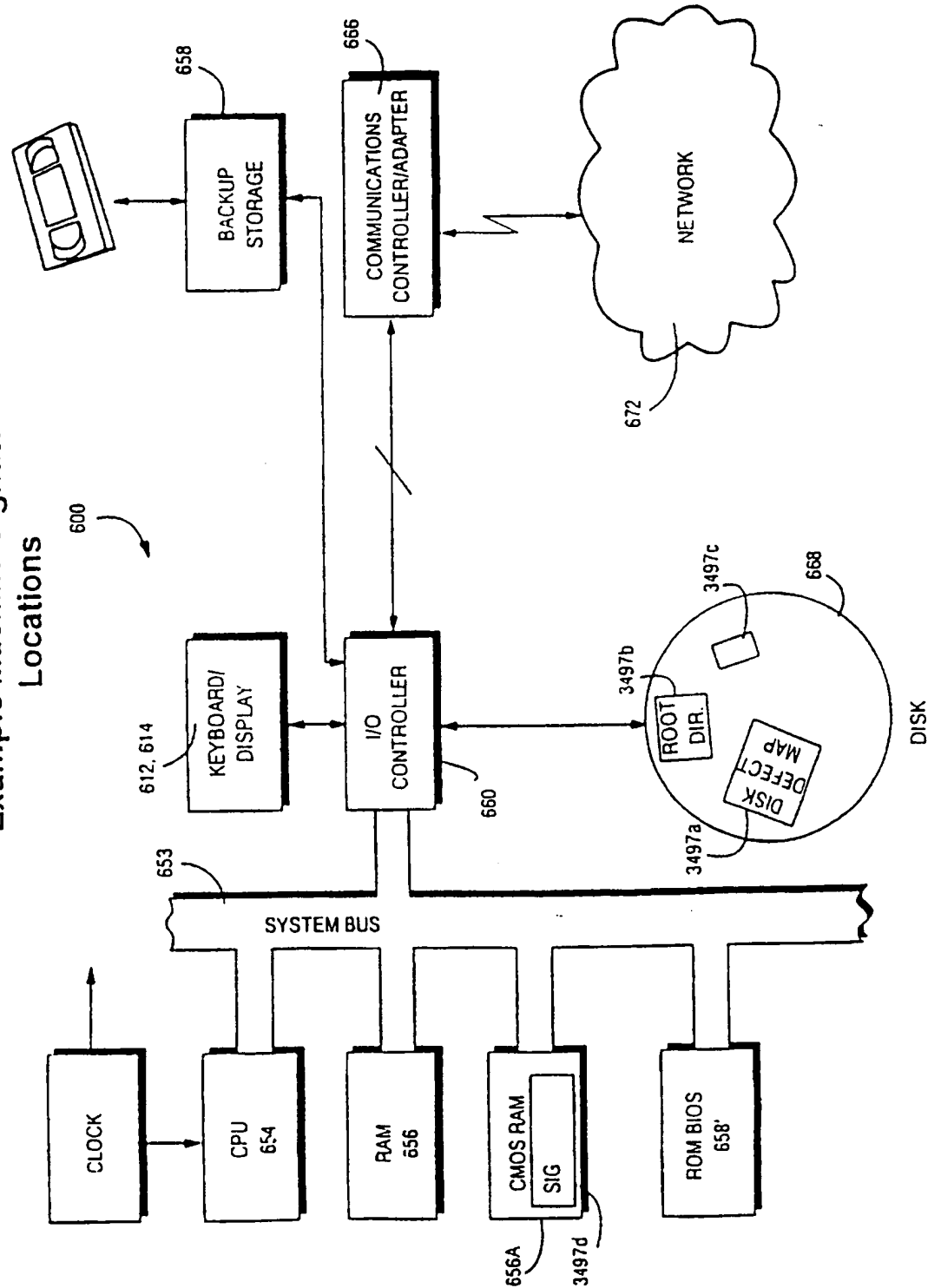
**Fig. 69E**  
Example Locations For Random  
Modifications and Fingerprints

**Fig. 69F**  
Example Customized Static  
Storage Layout



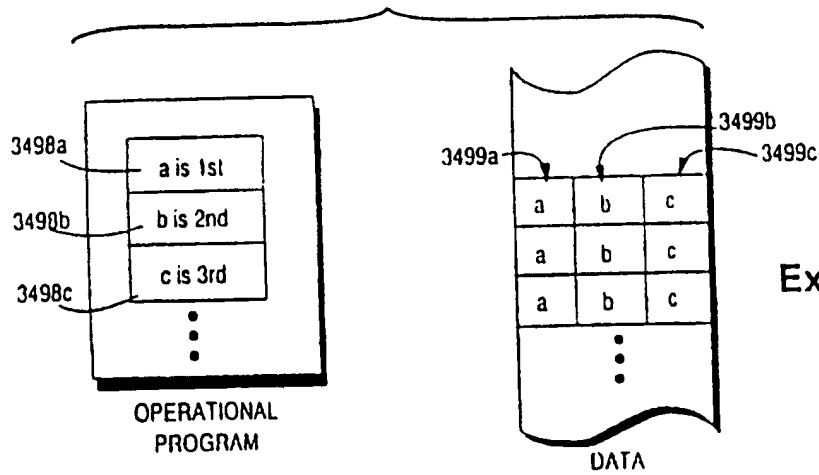
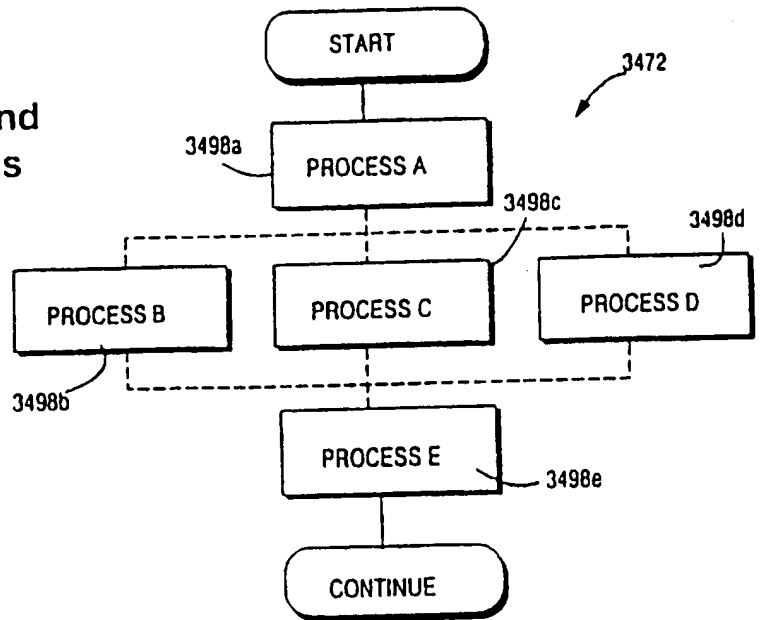
131/163

**Fig. 69G**  
Example Machine Signature  
Locations

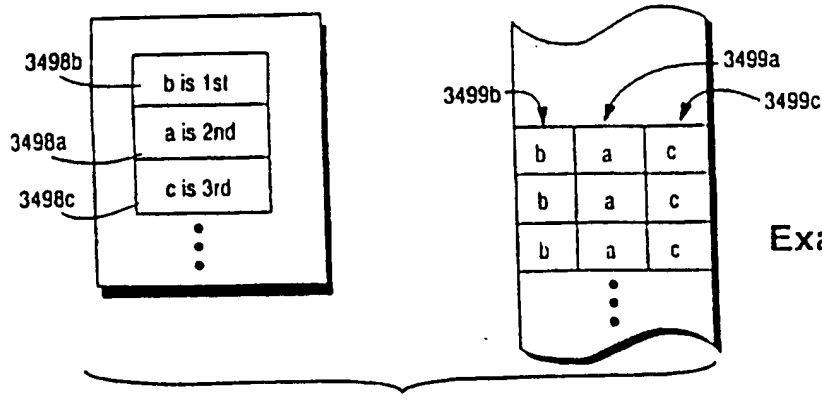


132/163

**Fig. 69H**  
Sequence Dependent and Independent Processes



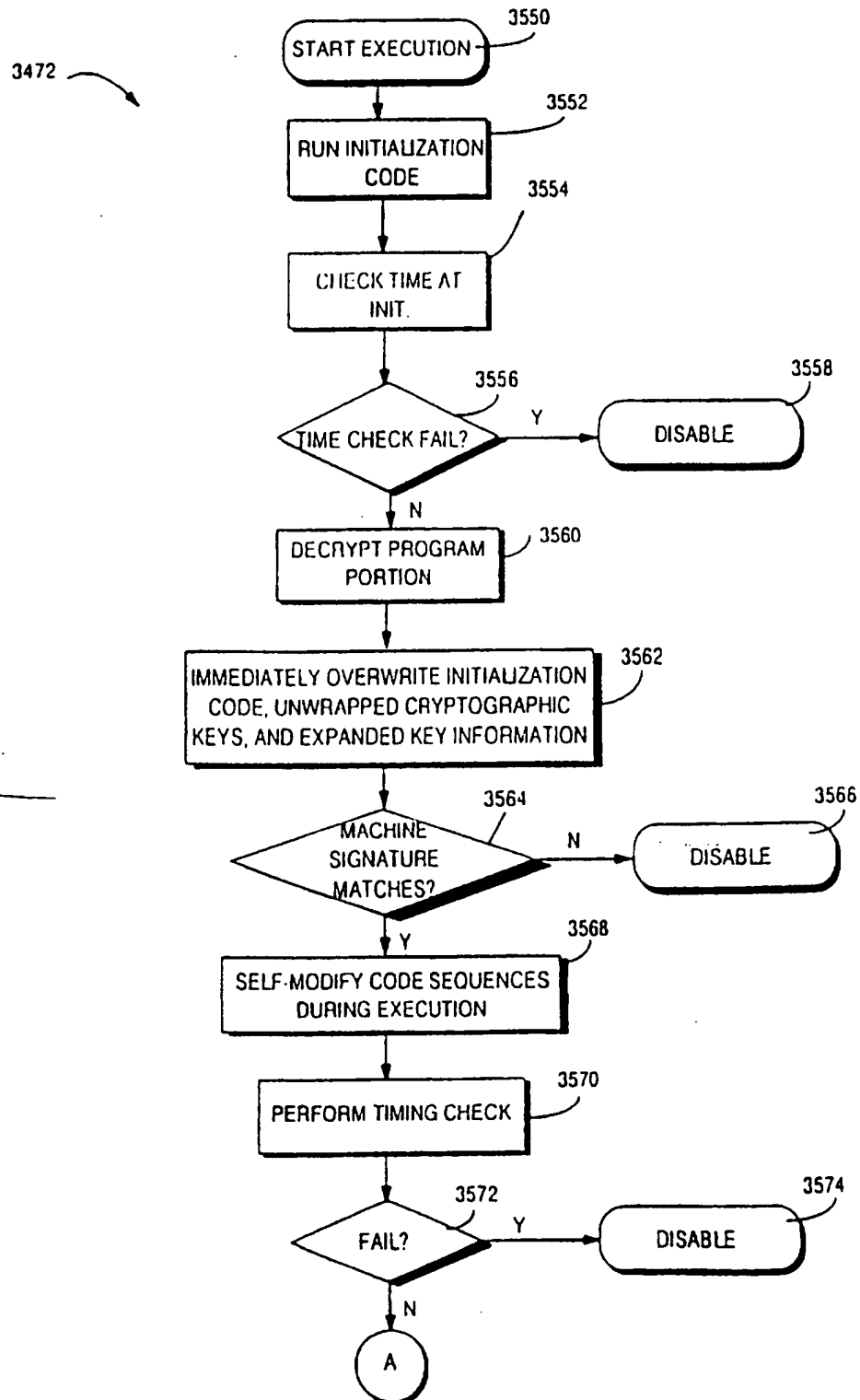
**Fig. 69I**  
Example First Order



**Fig. 69J**  
Example Second Order

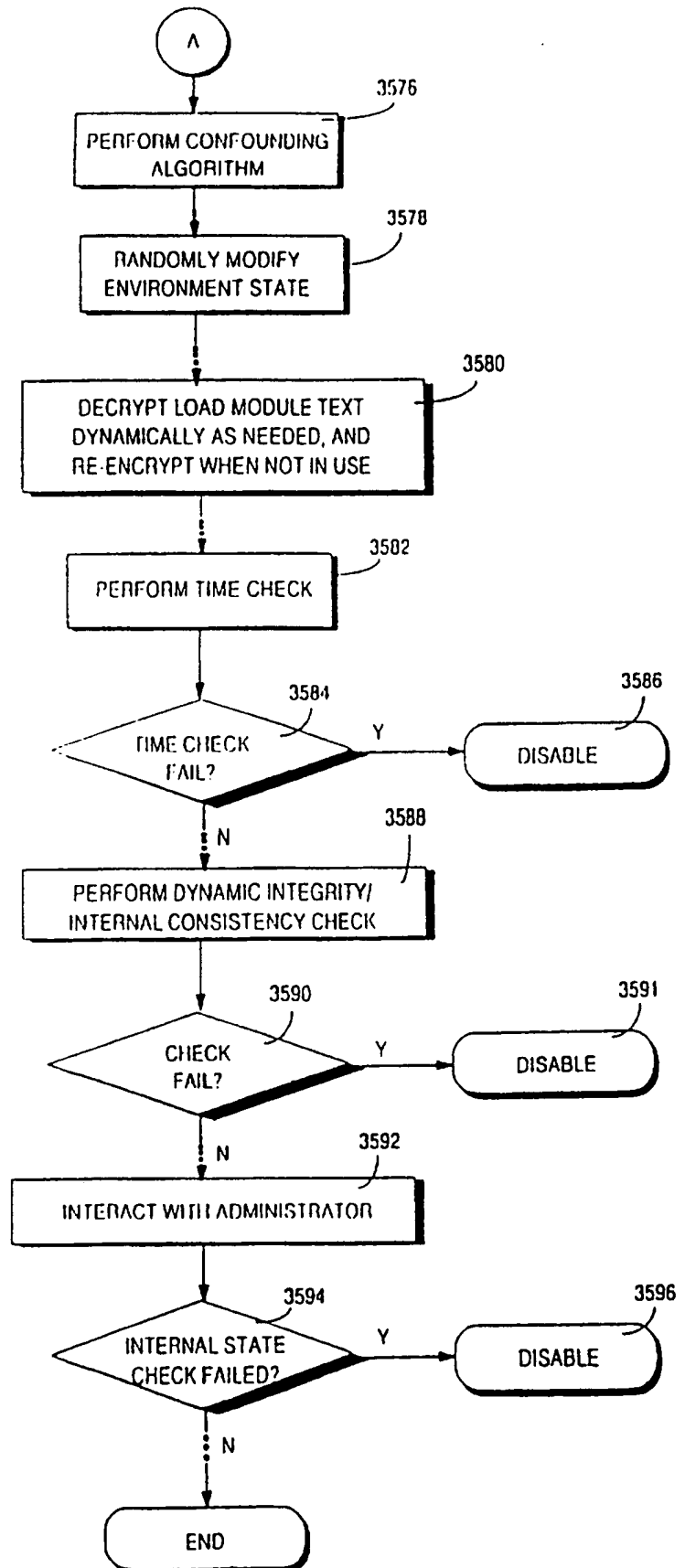
133/163

**Fig. 69K**  
Example Dynamic Protection  
Mechanisms



134/163

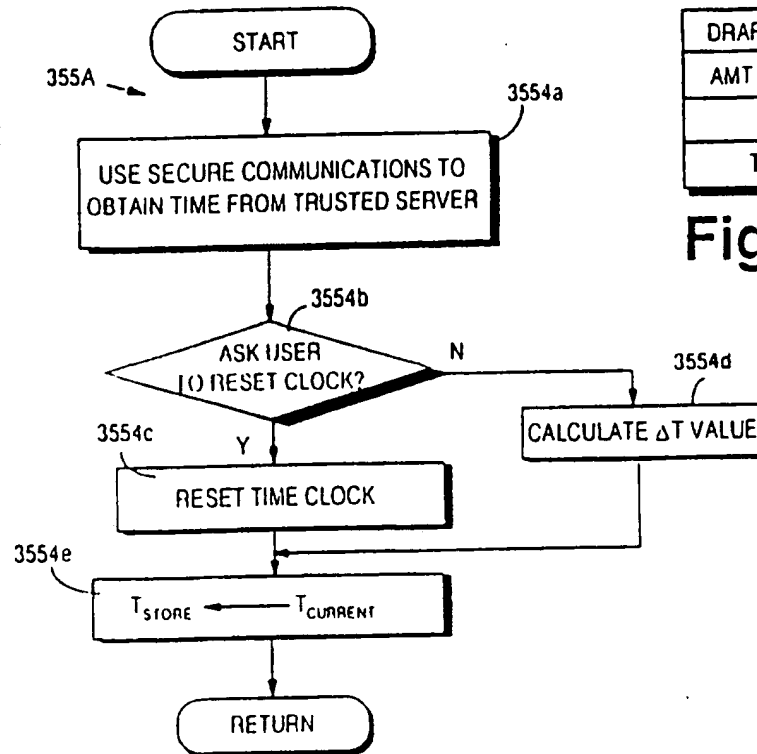
**Fig. 69L**  
Example Dynamic  
Protection  
Mechanisms





135/163

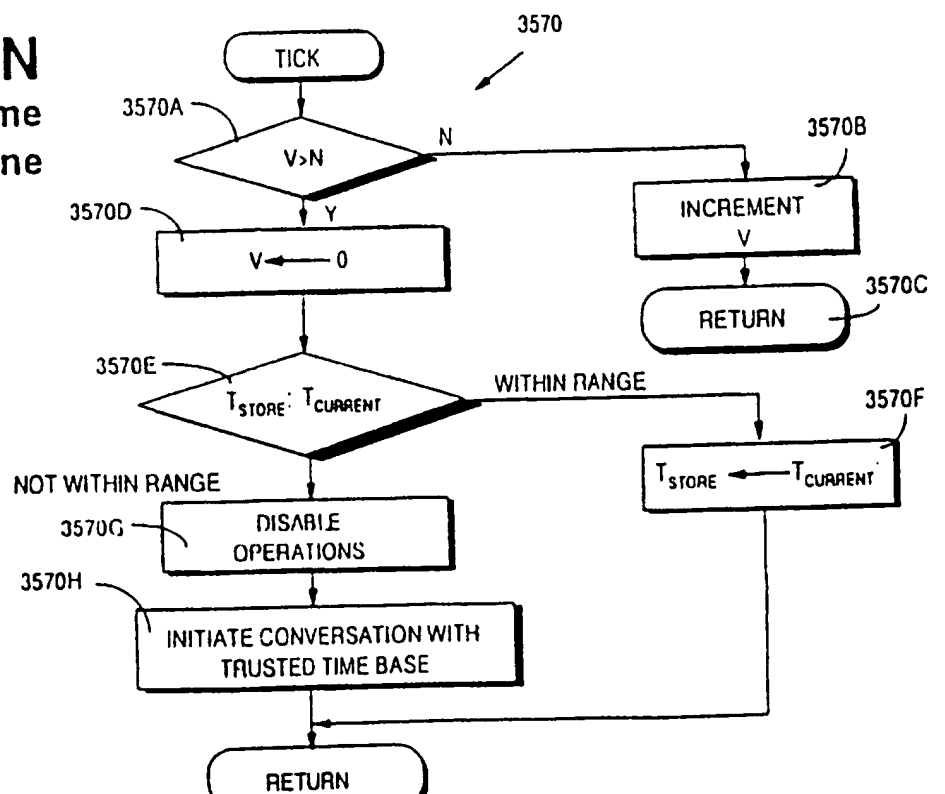
**Fig. 69M**  
Example Time  
Check At  
Initialization  
Routine



DRAFT BUDGET
AMT DRAFTED
$\Delta T$
$T_{STORE}$

**Fig. 69O**

**Fig. 69N**  
Example Time  
Check Routine



136/163

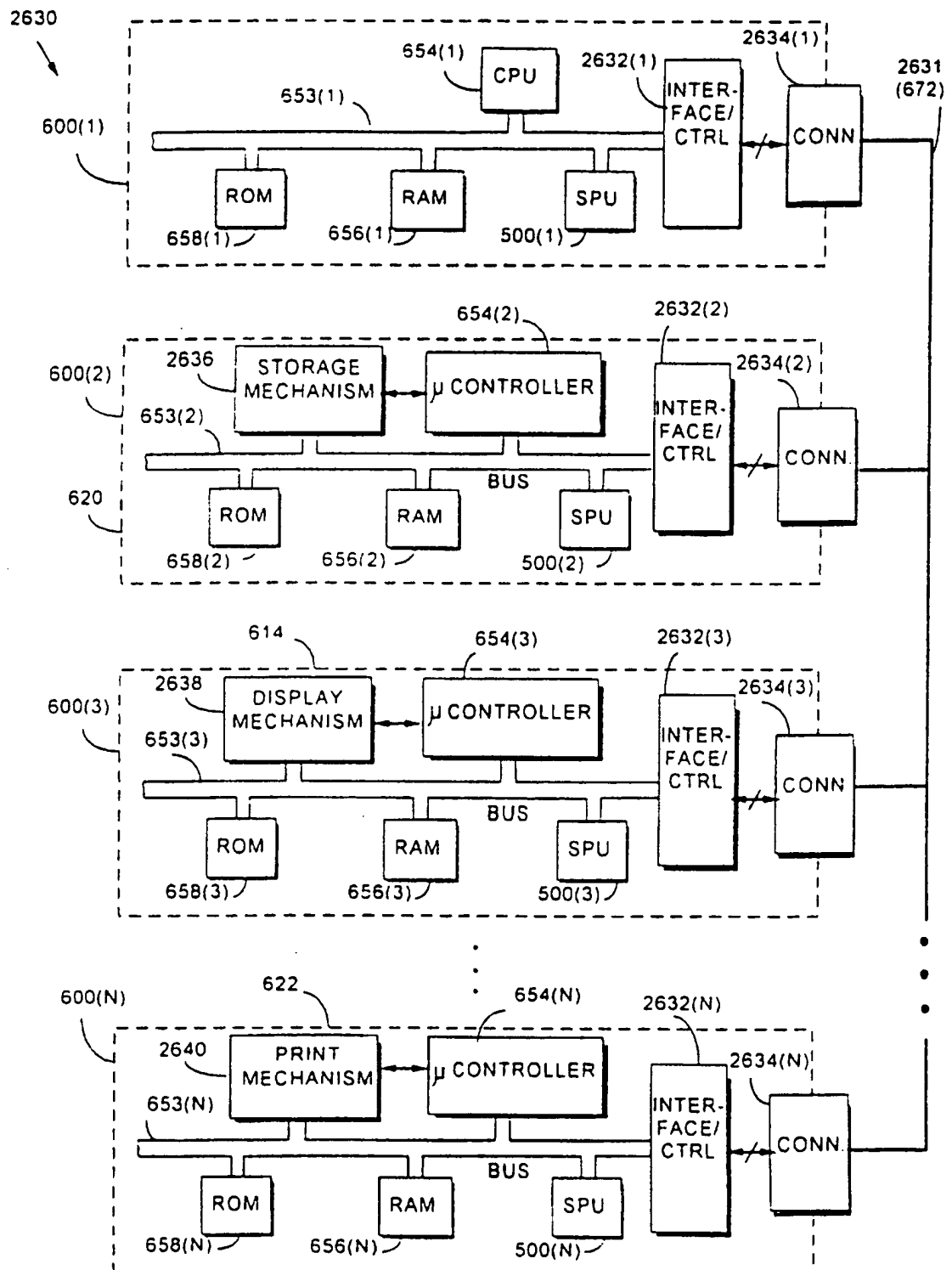
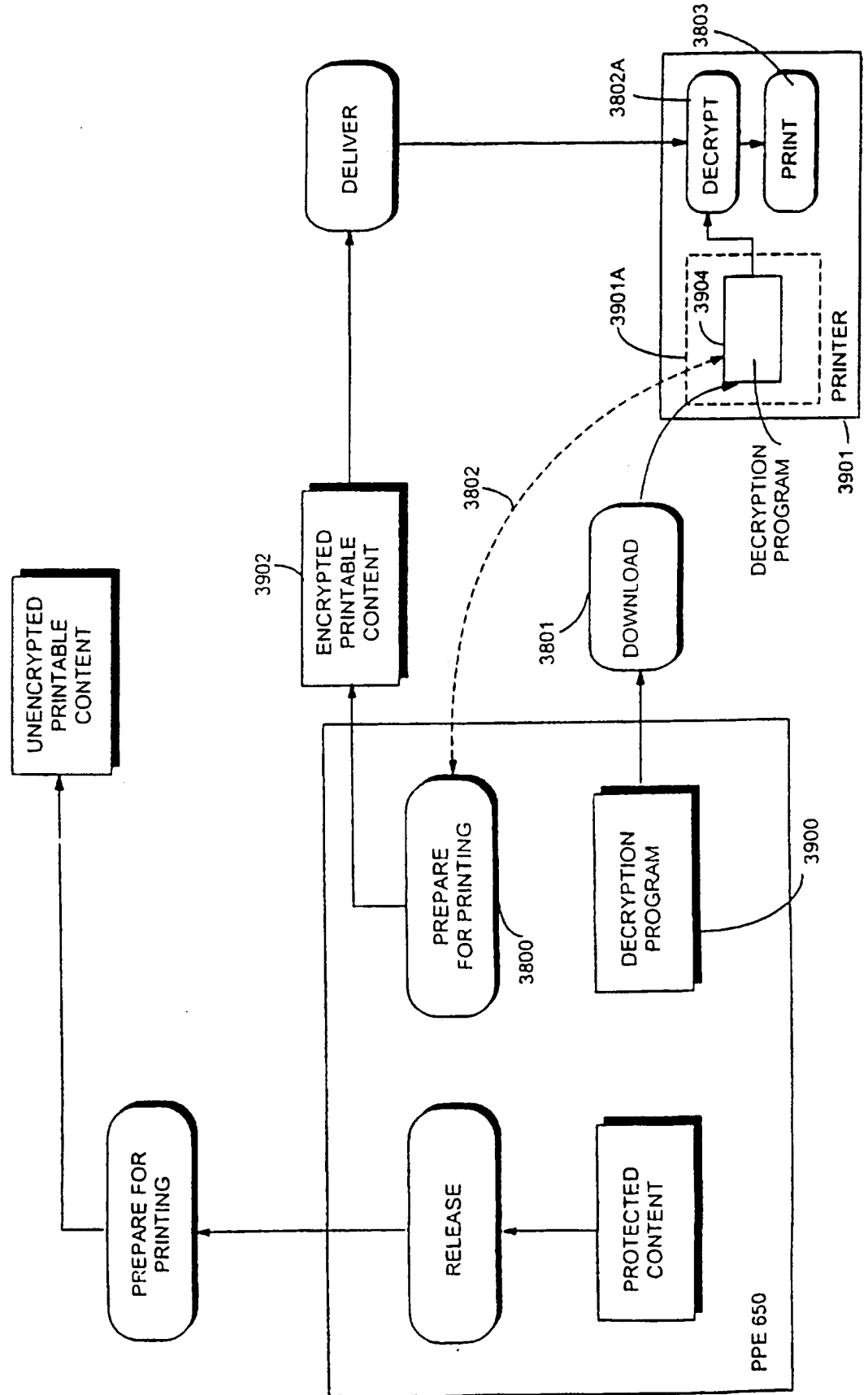


FIG. 70

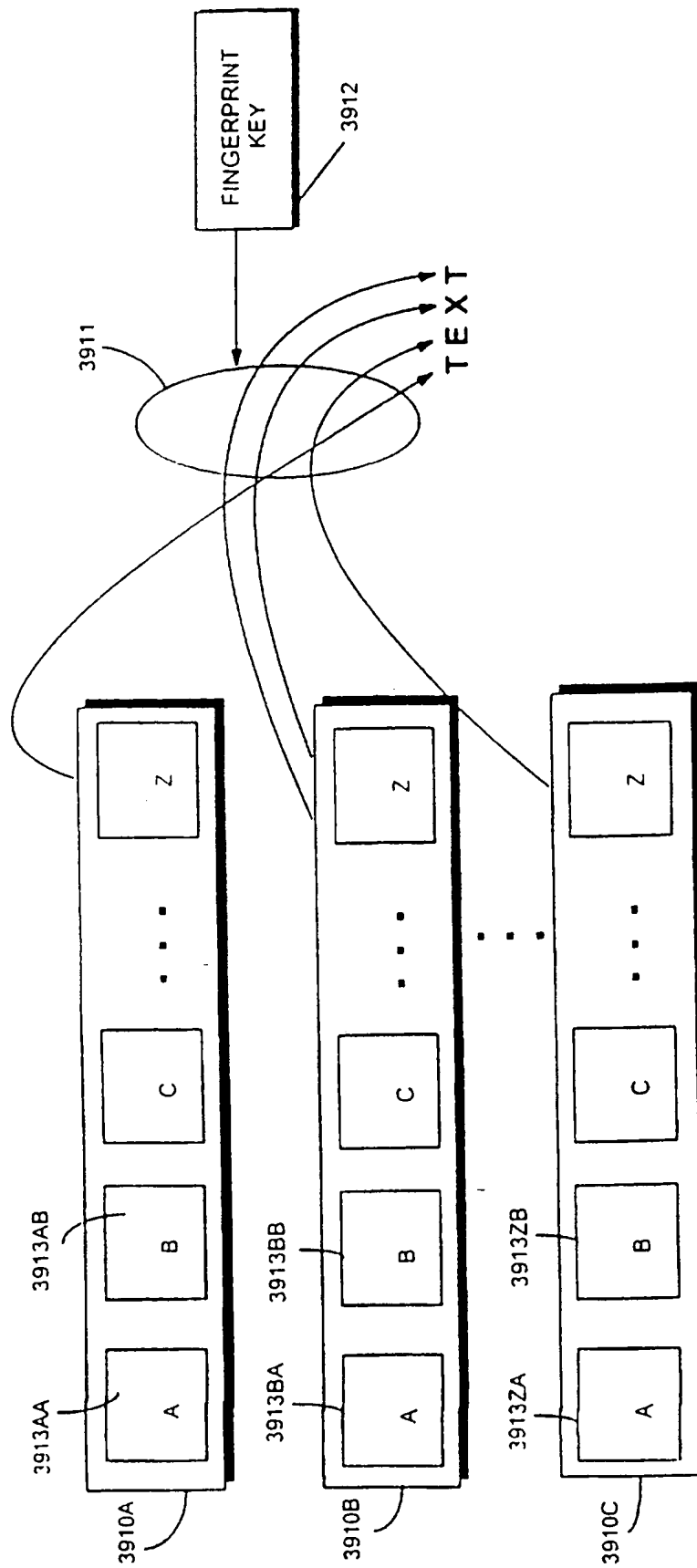
SUBSTITUTE SHEET (RULE 26)

**FIG. 70A**



138/163

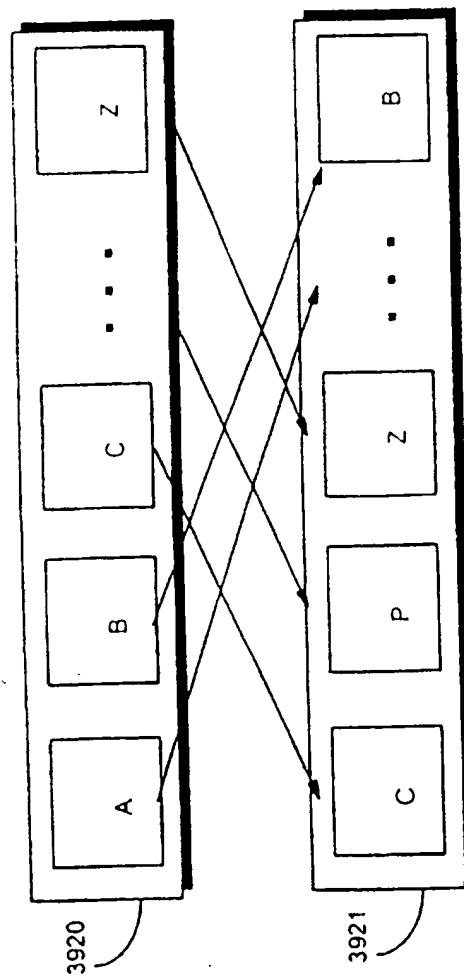
FIG. 70B



SUBSTITUTE SHEET (RULE 26)

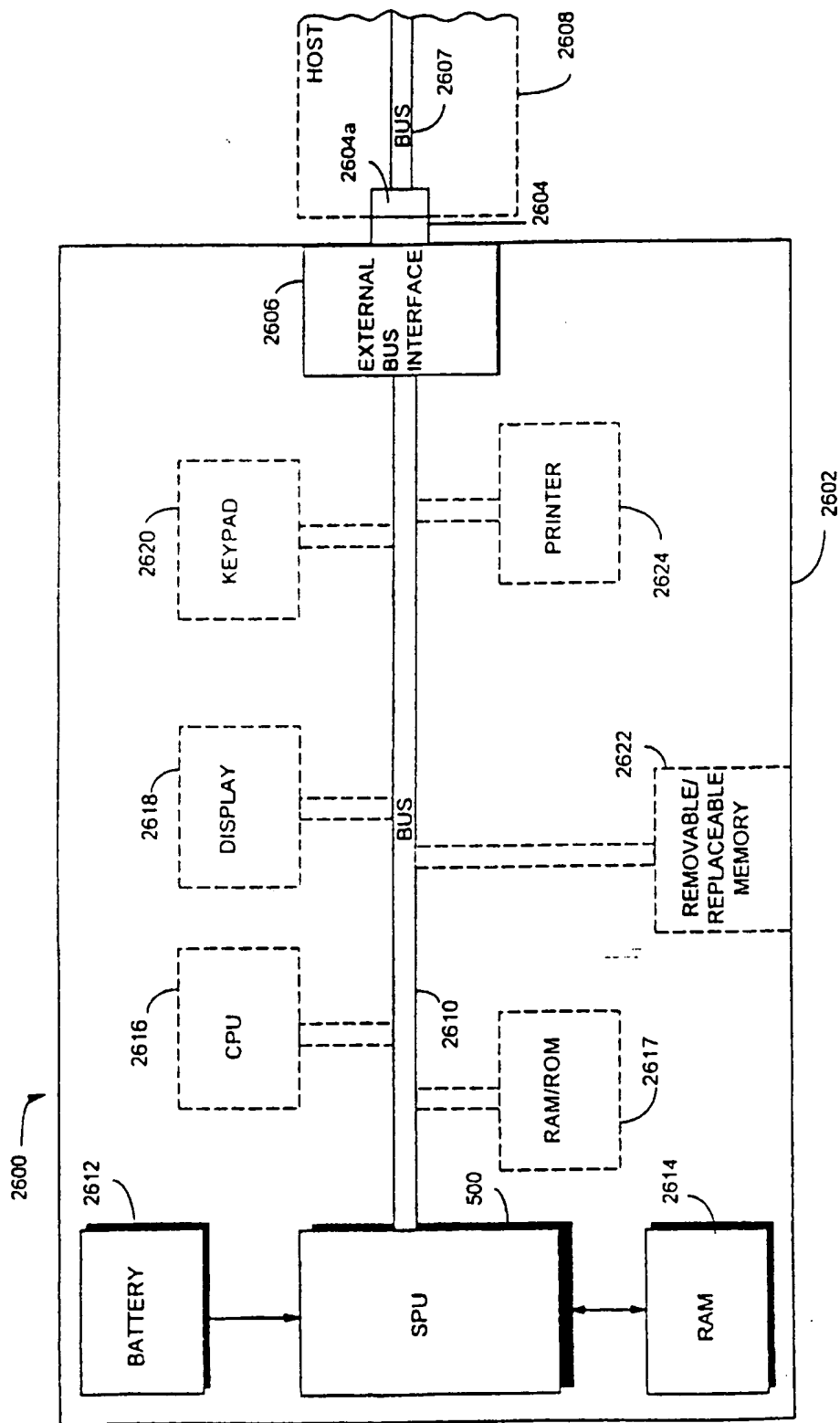
139/163

FIG. 70C



140/163

**FIG. 71**  
PORTABLE APPLIANCE



141/163

LOG IN USER INTERFACE

182

USER NAME: SHEAR. V.

PASSWORD: \* \* \* \* \*

☐ LOGIN AT STARTUP

LOGIN

CANCEL


HELP

FIG. 72A

FIG. 72B

2660

YOU HAVE REQUESTED THESE PROPERTIES:



LOONEY TUNES NEWS!

2662

PROPERTY INFO

Your Cost: \$7.50

APPROVE

SUSPEND

CANCEL

MORE OPTIONS

2664

142/163

FIG. 72C

**SET LIMITS:**

SESSION DOLLAR LIMIT \$  2666

TRANSACTION DOLLAR LIMIT: \$  2668

TIME LIMIT (IN MINUTES)  2670

UNIT LIMIT  2672

2674



143/163

FIG. 72D

**YOU HAVE REQUESTED THESE PROPERTIES:**

**LOONEY TUNE NEWS!**

**YOUR COST : \$7.50**

**PROPERTY INFO**

**More Options ☒ Show Thumbnail**

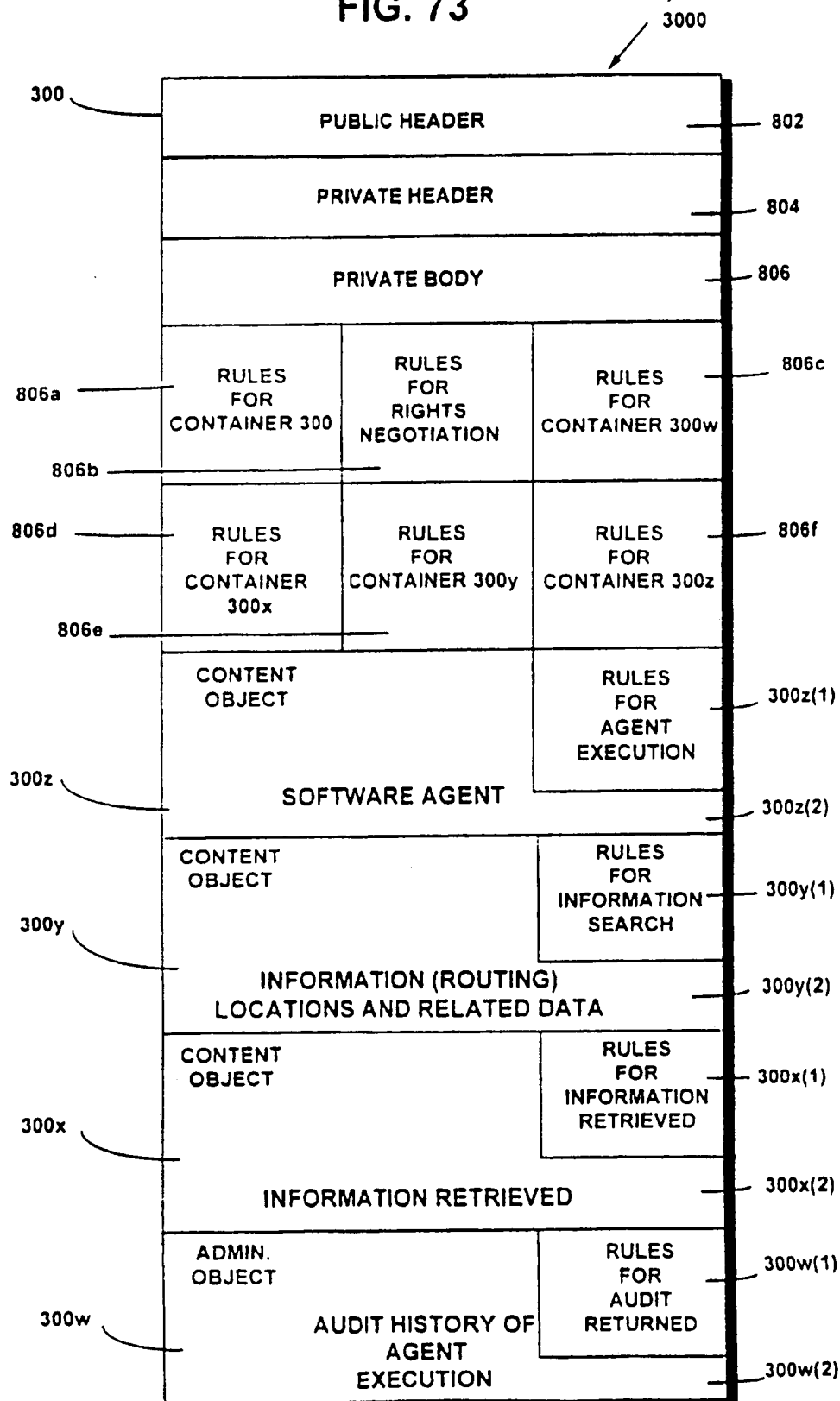
**CANCEL** **SUSPEND** **APPROVE**

PROPERTY:	SIZE:	PUBLISHER:	AMOUNT:	UNITS:	COST/UNIT:	TYPE:	USE?	LINKS:	HIST:
CHUCK JONES BIOGRA...	256KB	WARNER NEW MEDIA	64	KBYTE	\$1 25	PREVIEW	✓	●	
▼ BUGS BUNNY .JPE...	1MB	WARNER NEW MEDIA	1	RECORD	\$5 00	DISPLAY	✓	●	
BUGS BUNNY .JPE...	1MB	WARNER NEW MEDIA	10	RECORD	\$3 50	DISPLAY		●	
BUGS BUNNY .JPE...	1MB	WARNER NEW MEDIA	25	RECORD	\$2 50	DISPLAY		●	
FRIZ FRELENG BIOGRA...	256KB	WARNER NEW MEDIA	120	SECTOR	\$5 00	PRINT		●	
TEX AVERY BIOGRAP	256KB	WARNER NEW MEDIA	50	PERCENT	\$2 50	COPY		●	
► DUCK! RABBITI DU...	64MB	WARNER NEW MEDIA	7 0	MINUTE	\$7 50	COPY.PRO		●	
MEL BLANC BIOGRAPH	256KB	WARNER NEW MEDIA	1	SPECIAL	\$25 25	INSTALL		●	
LOONEY TUNES DATAB	600MB	WARNER NEW MEDIA	1	OBJECT	\$2000 00	ALL		●	

**SET LIMITS...** **SHOW BUDGETS** **ACQUIRE BUDGET...** **HISTORY...** **TRANSFER...** **PREFERENCES...** **FEEDBACK...** **HELP!**

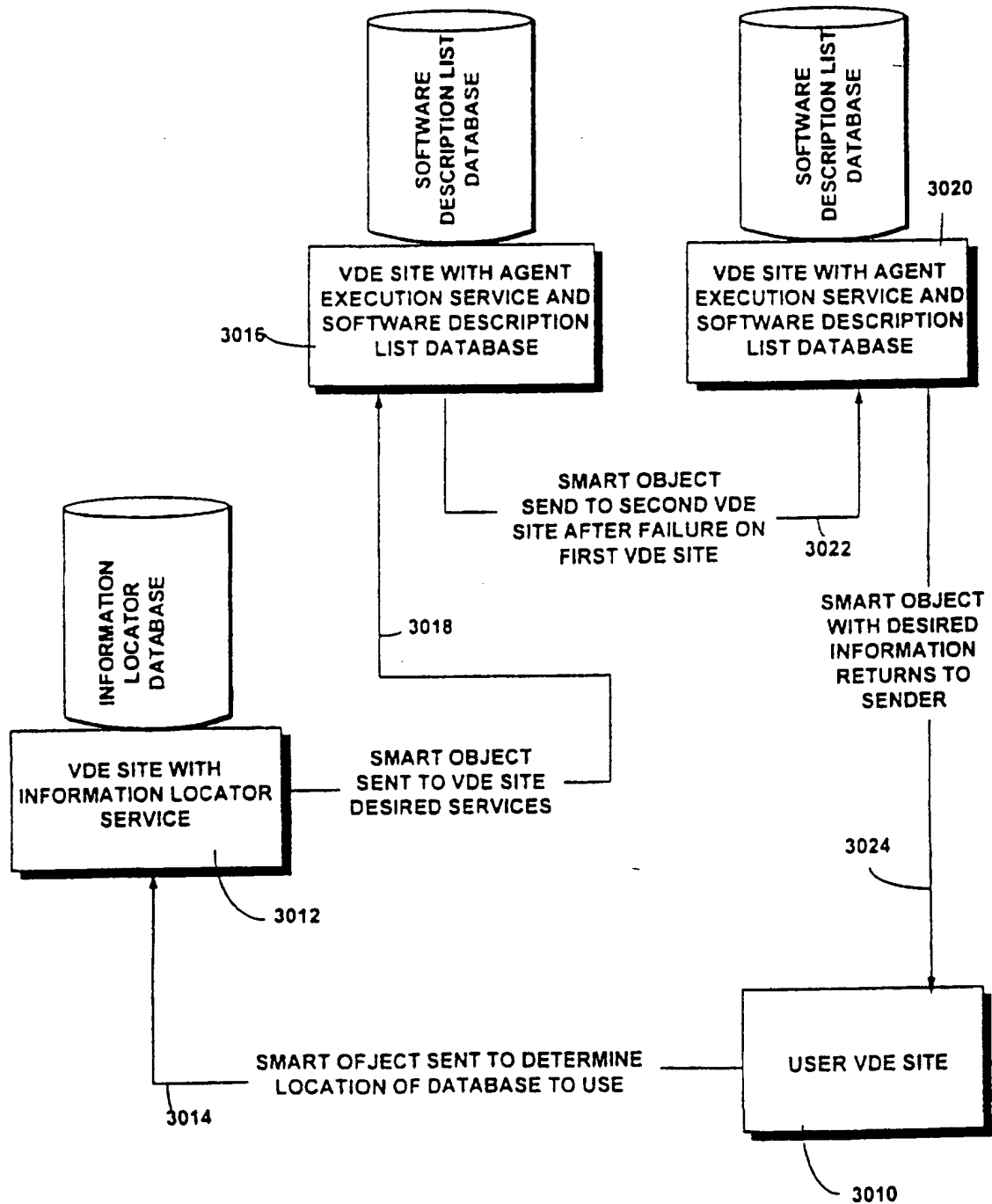
144/163

FIG. 73



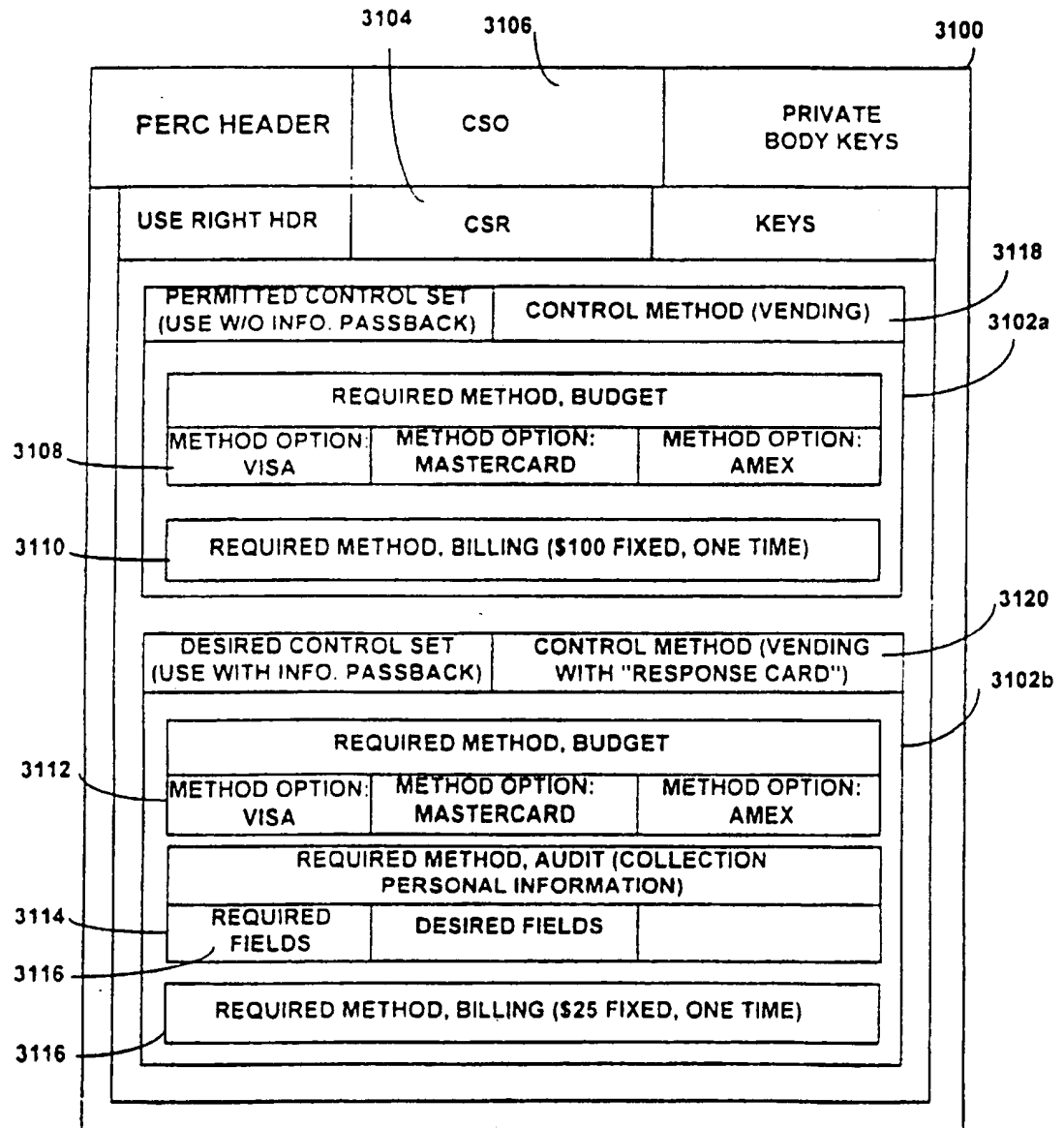
145/163

FIG. 74



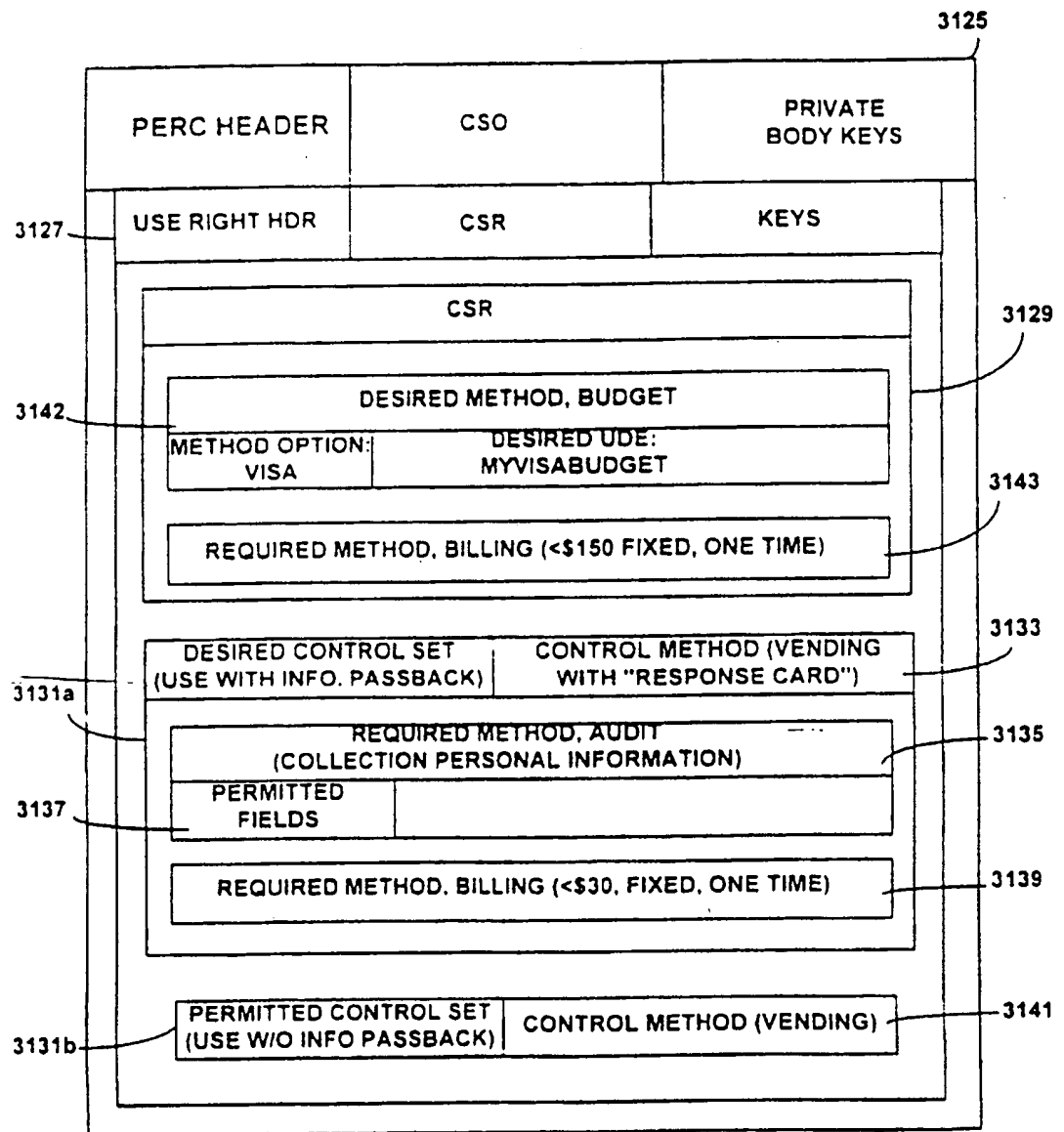
146/163

FIG. 75A



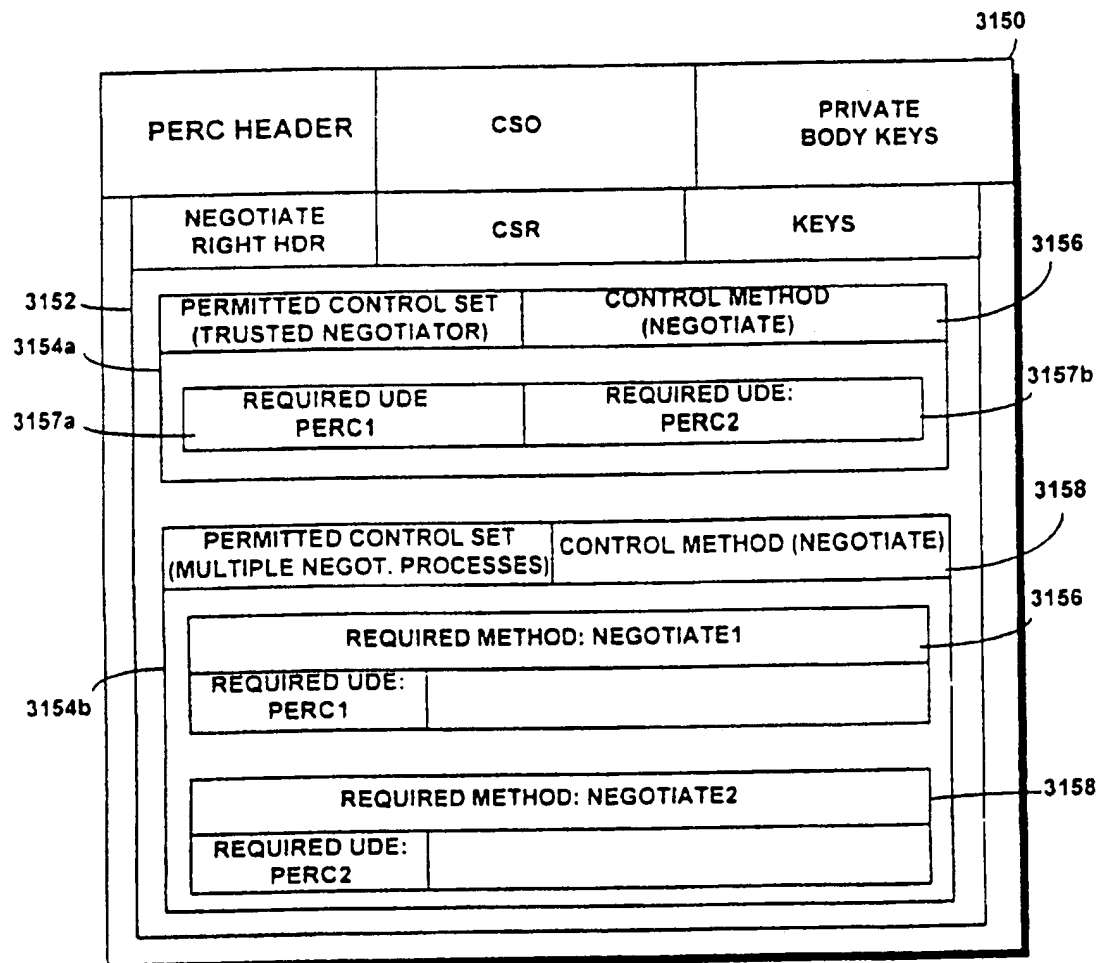
147/163

FIG. 75B



148/163

FIG. 75C



149/163

FIG. 75D

URT HEADER		CSO		DIGITAL SIGNATURE	
USE RIGHT HDR		CSR			
CONTROL SET(USE WITH INFO. PASSBACK)		CONTROL METHOD(VENDING WITH "RESPONSE CARD")			
REQUIRED METHOD, BUDGET					
METHOD OPTION: VISA		DESIRED UDE: MYVISABUDGET			
REQUIRED METHOD, AUDIT (COLLECTION PERSONAL INFORMATION)					
PERMITTED FIELDS					
REQUIRED METHOD, BILLING(\$25. FIXED, ONE TIME)					

3160

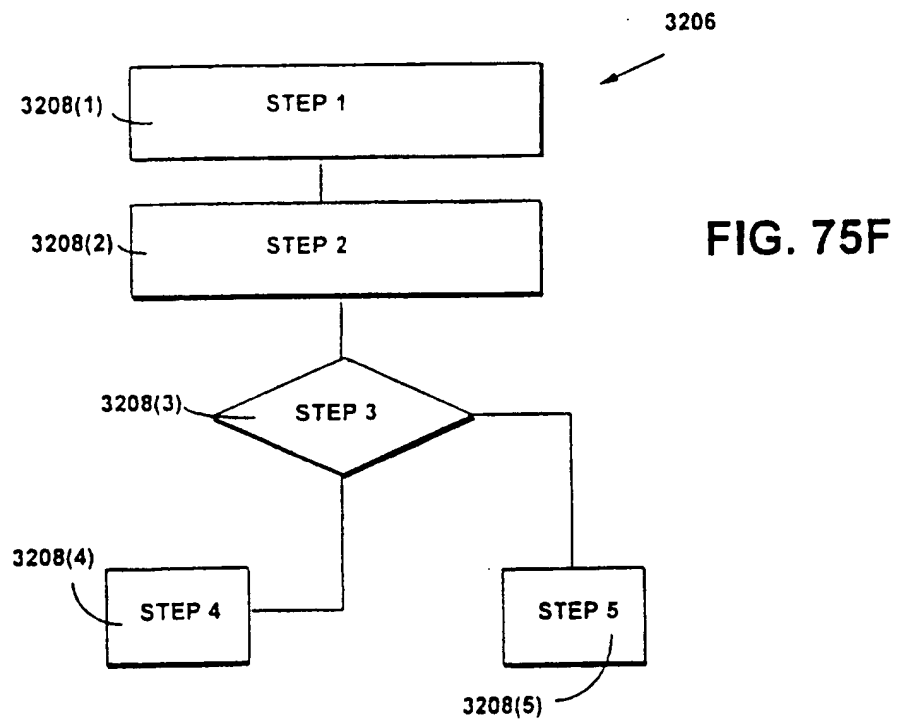
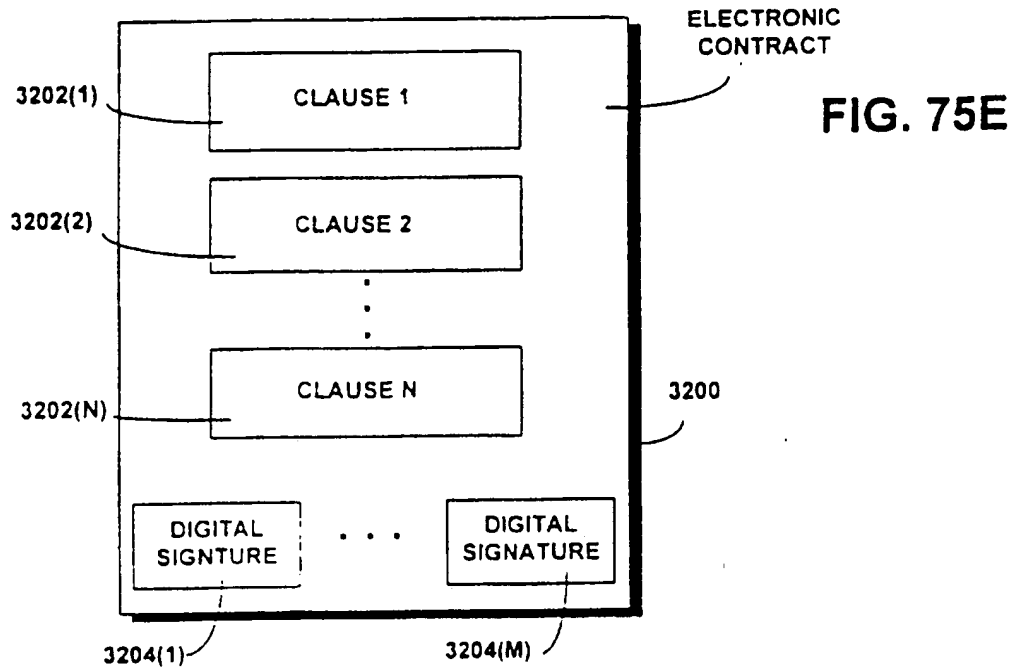
3162

3164

3166

3170

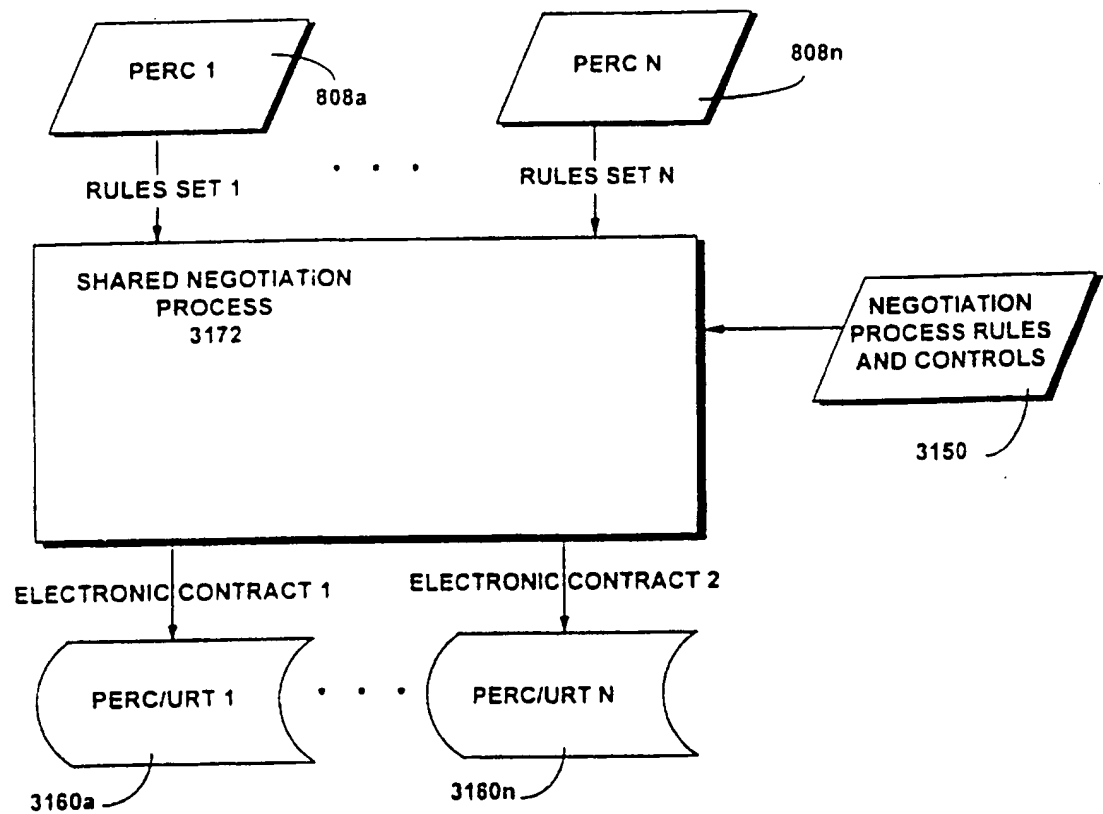
150/163





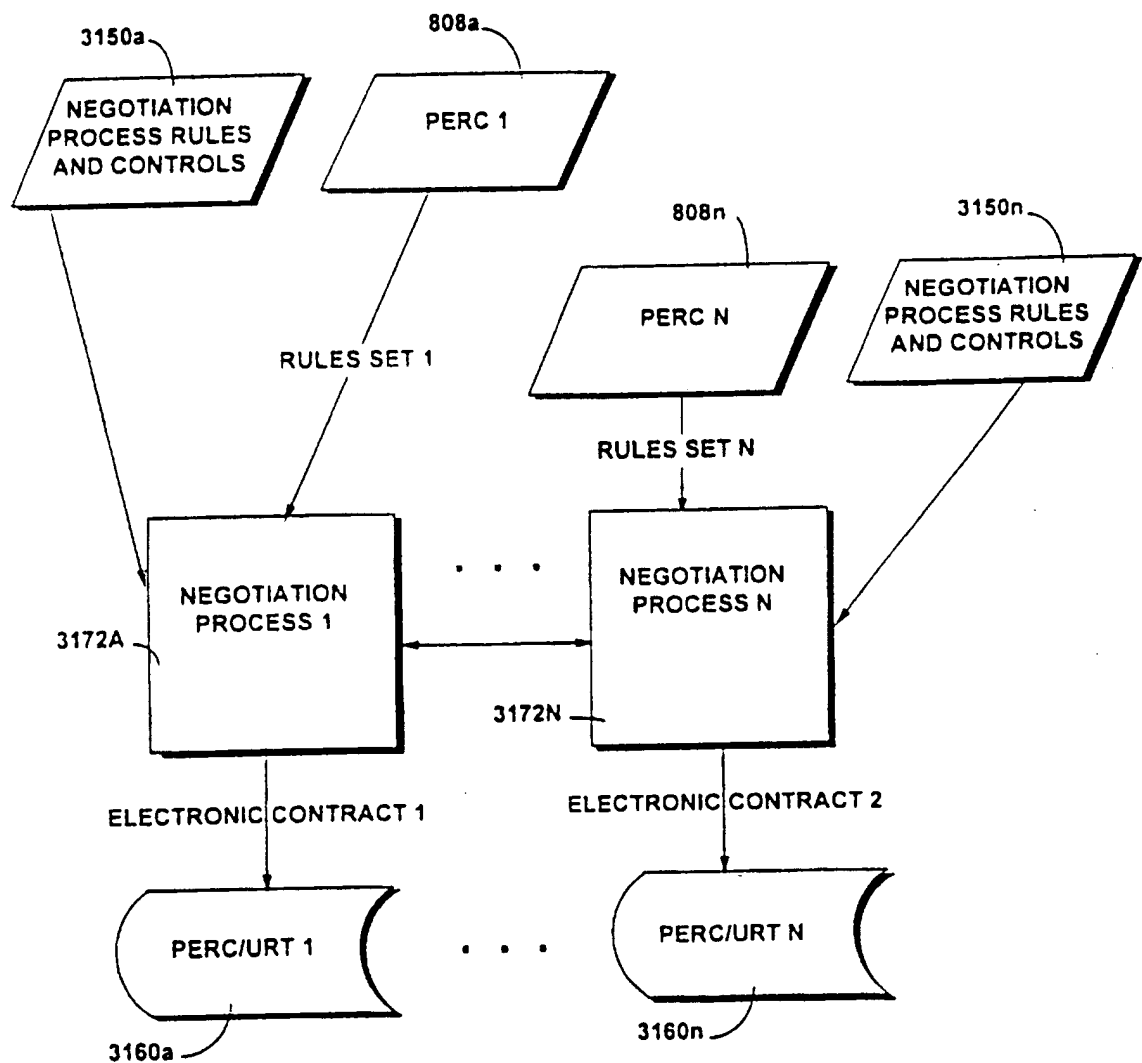
151/163

FIG. 76A



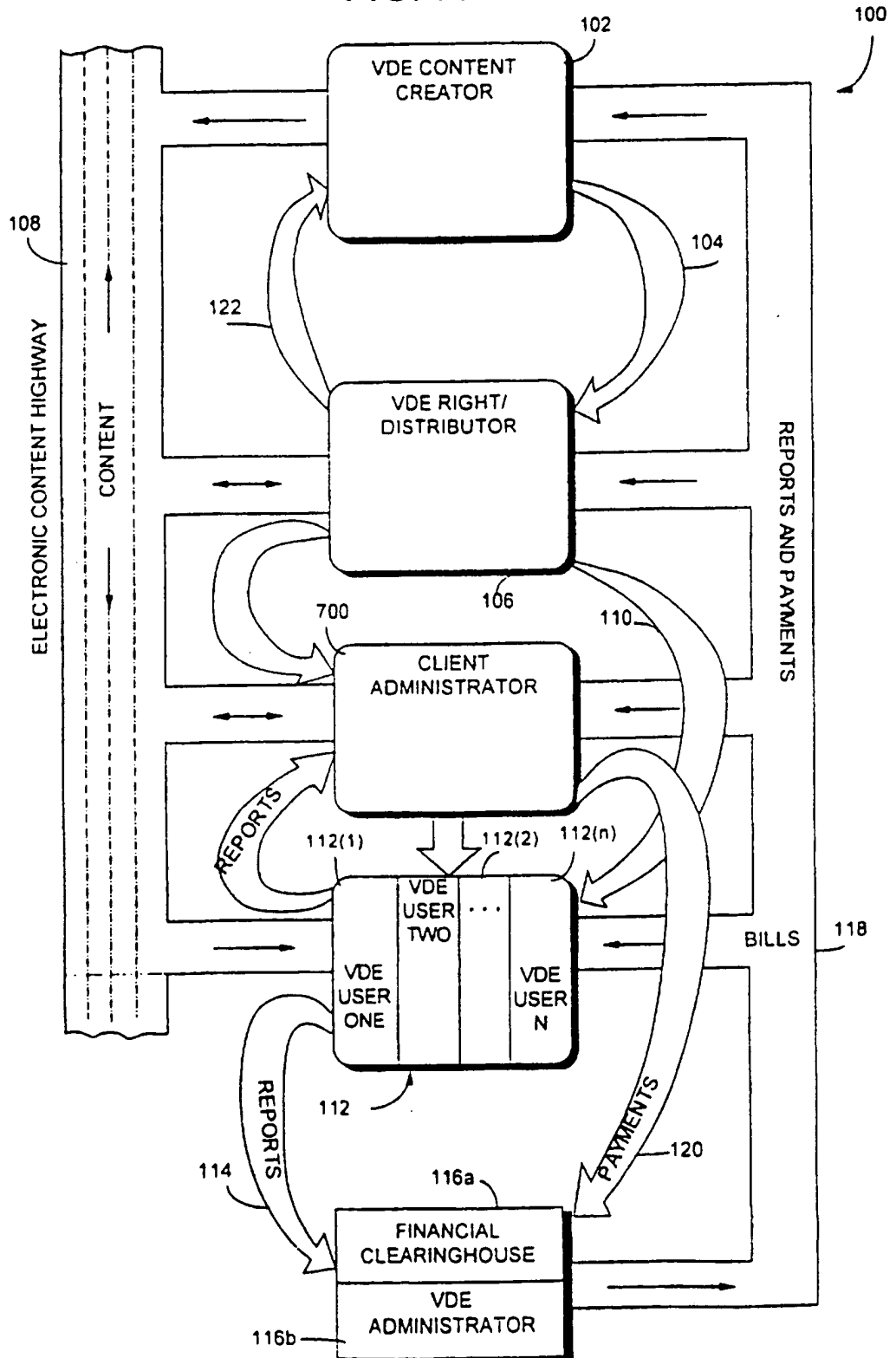
152/163

FIG. 76B



153/163

FIG. 77



154/163

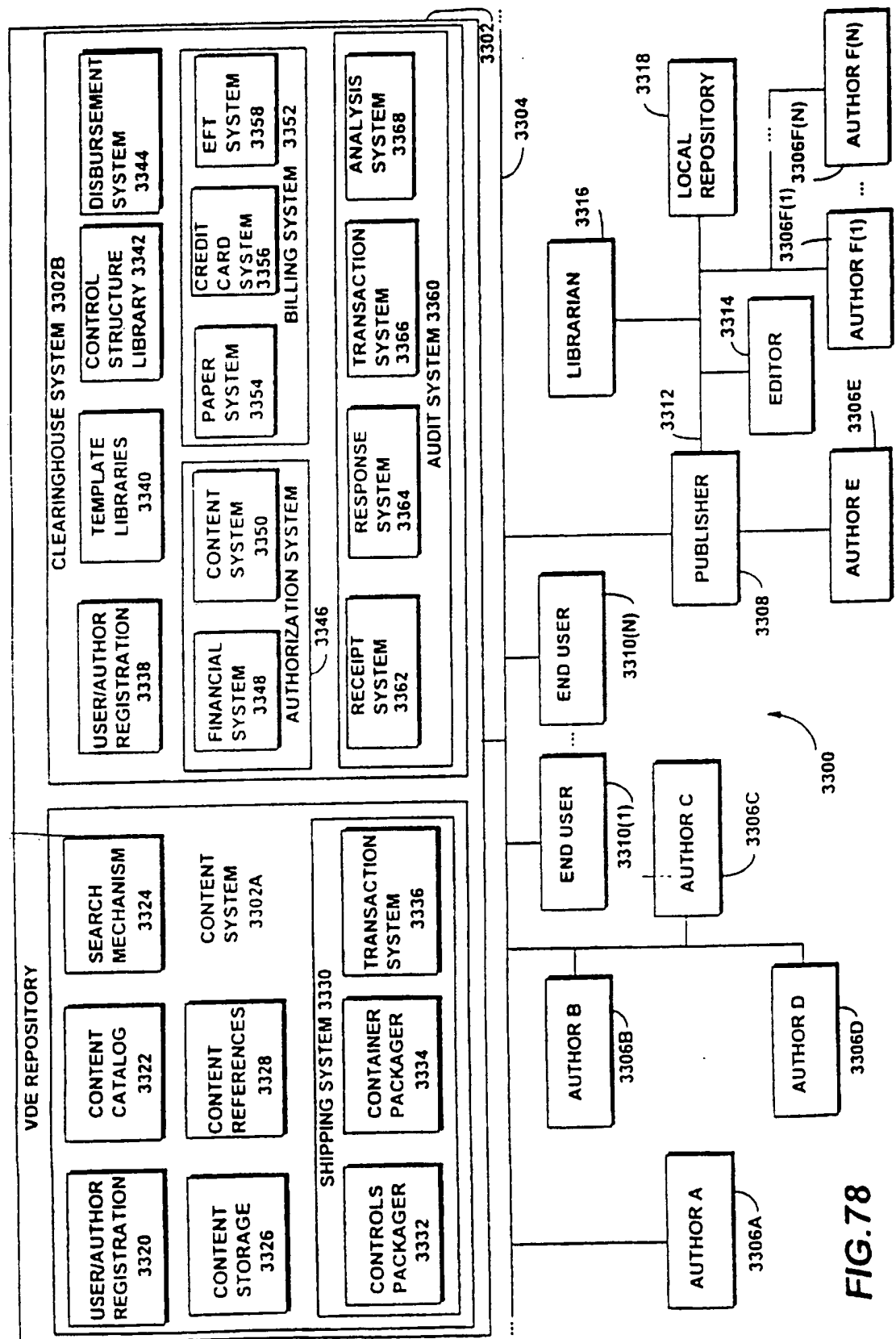
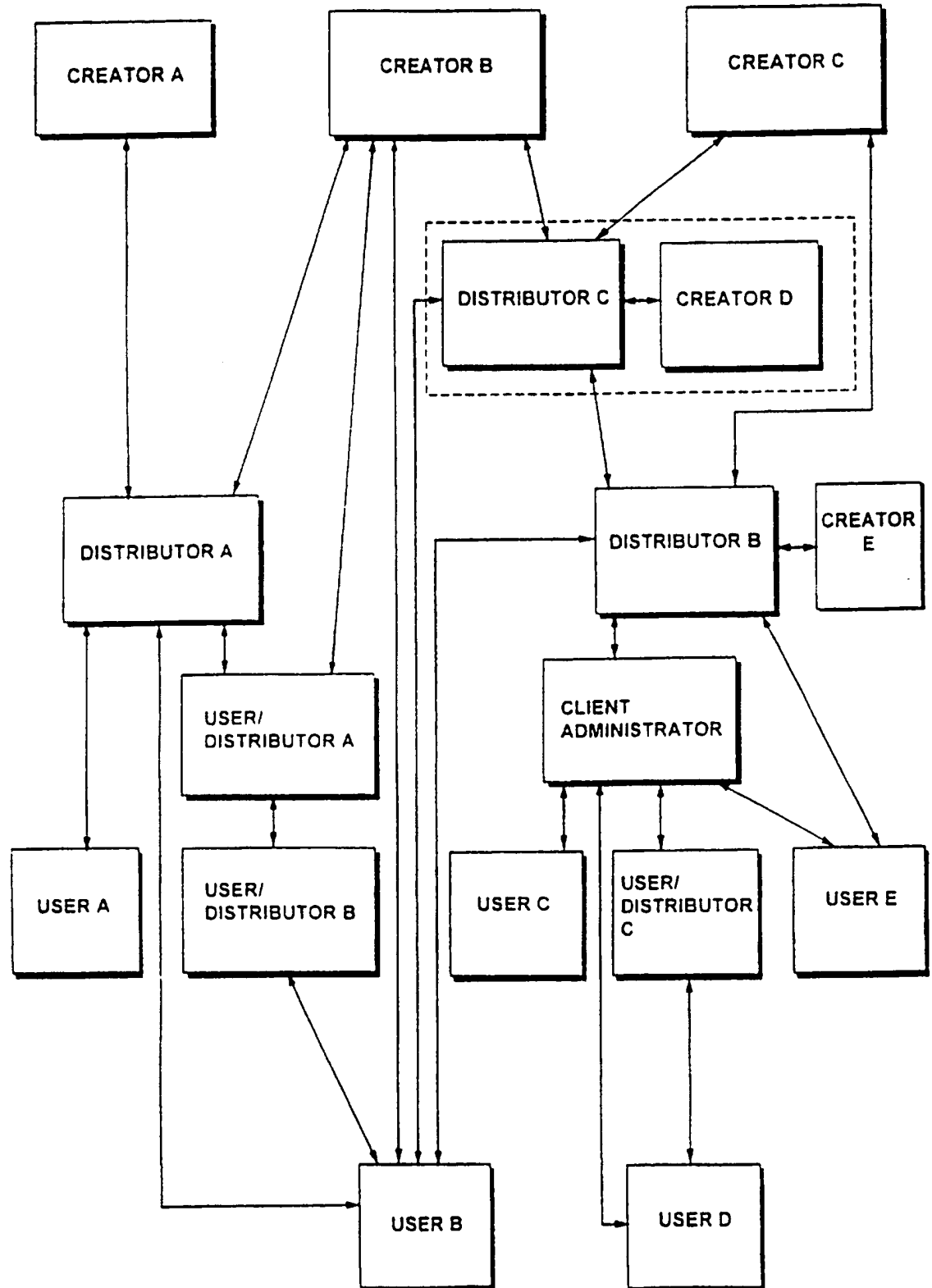


FIG.78

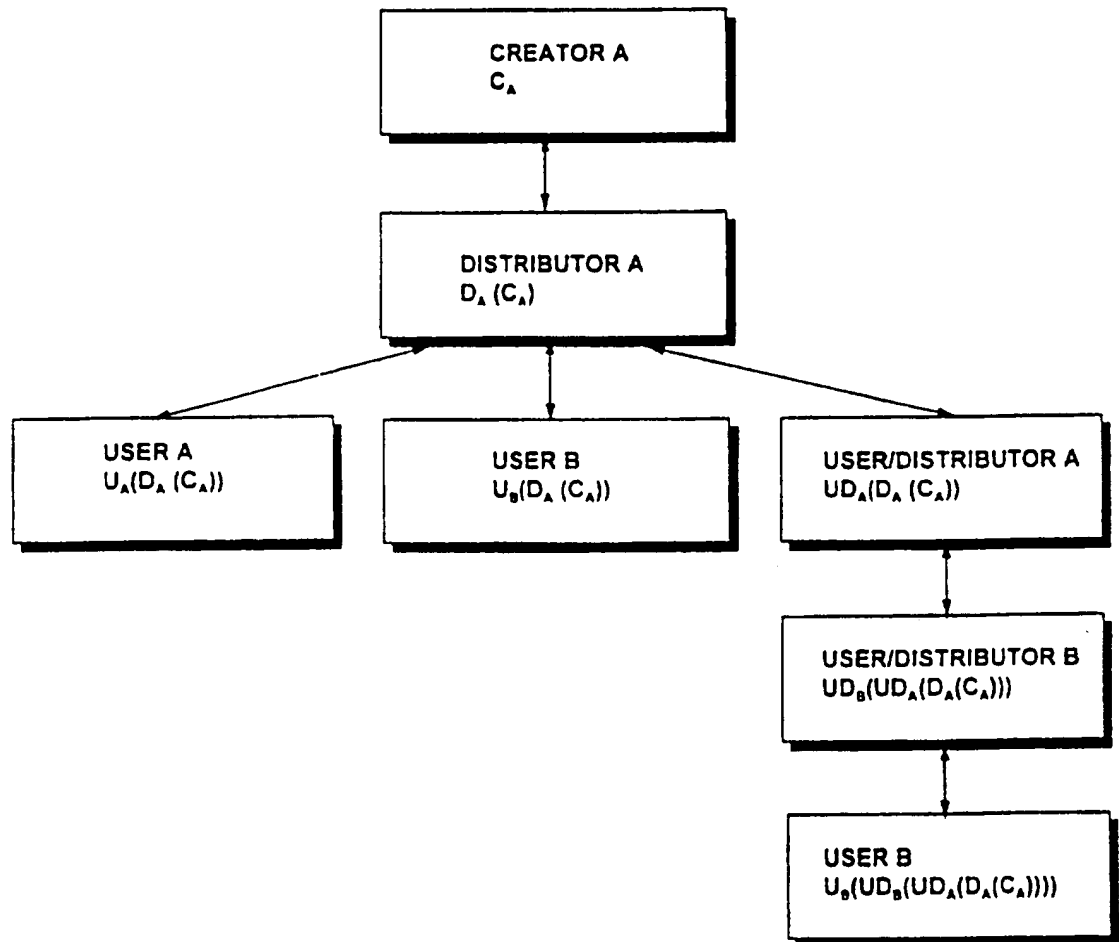
155/163

FIG. 79



156/163

FIG. 80



157/163

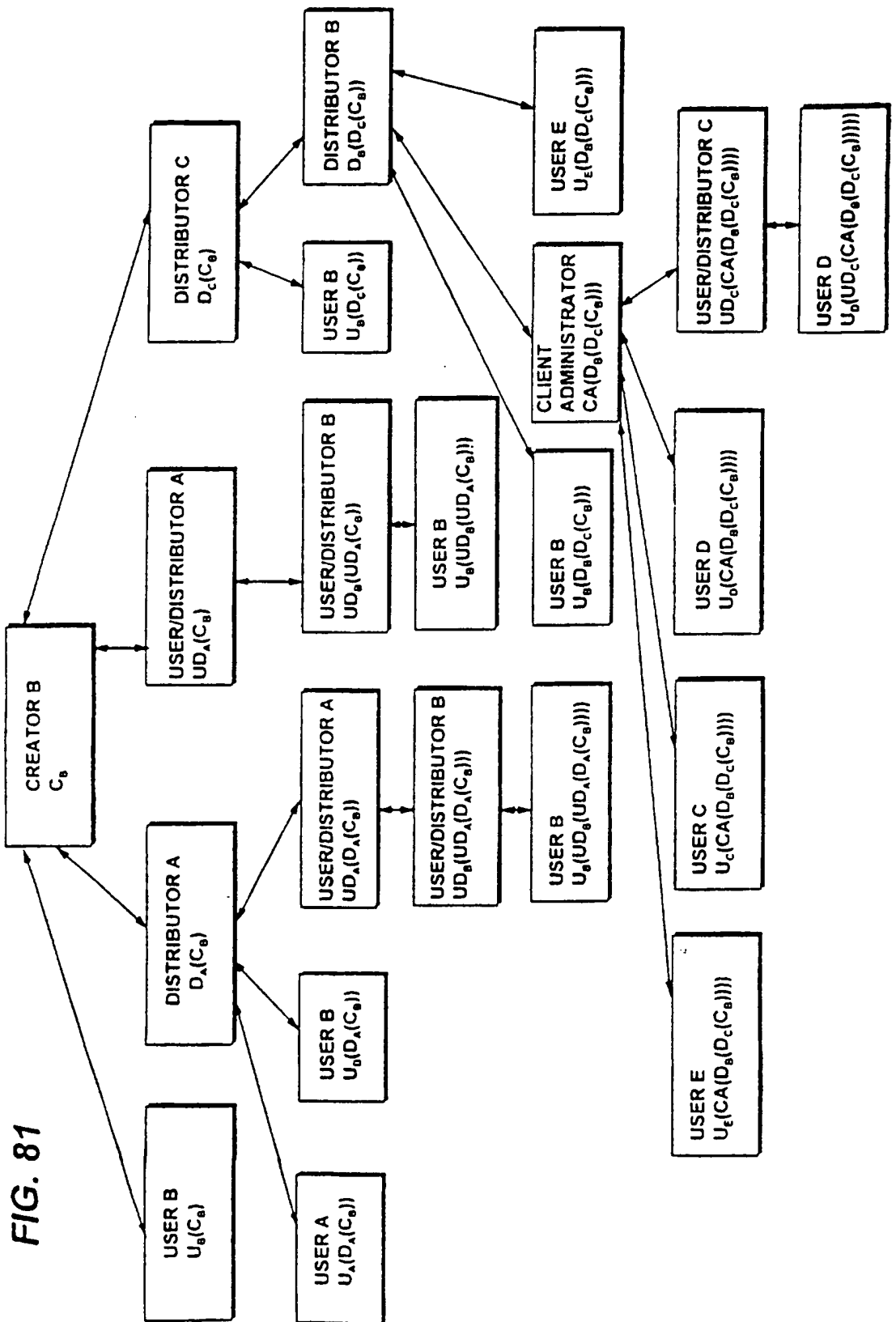
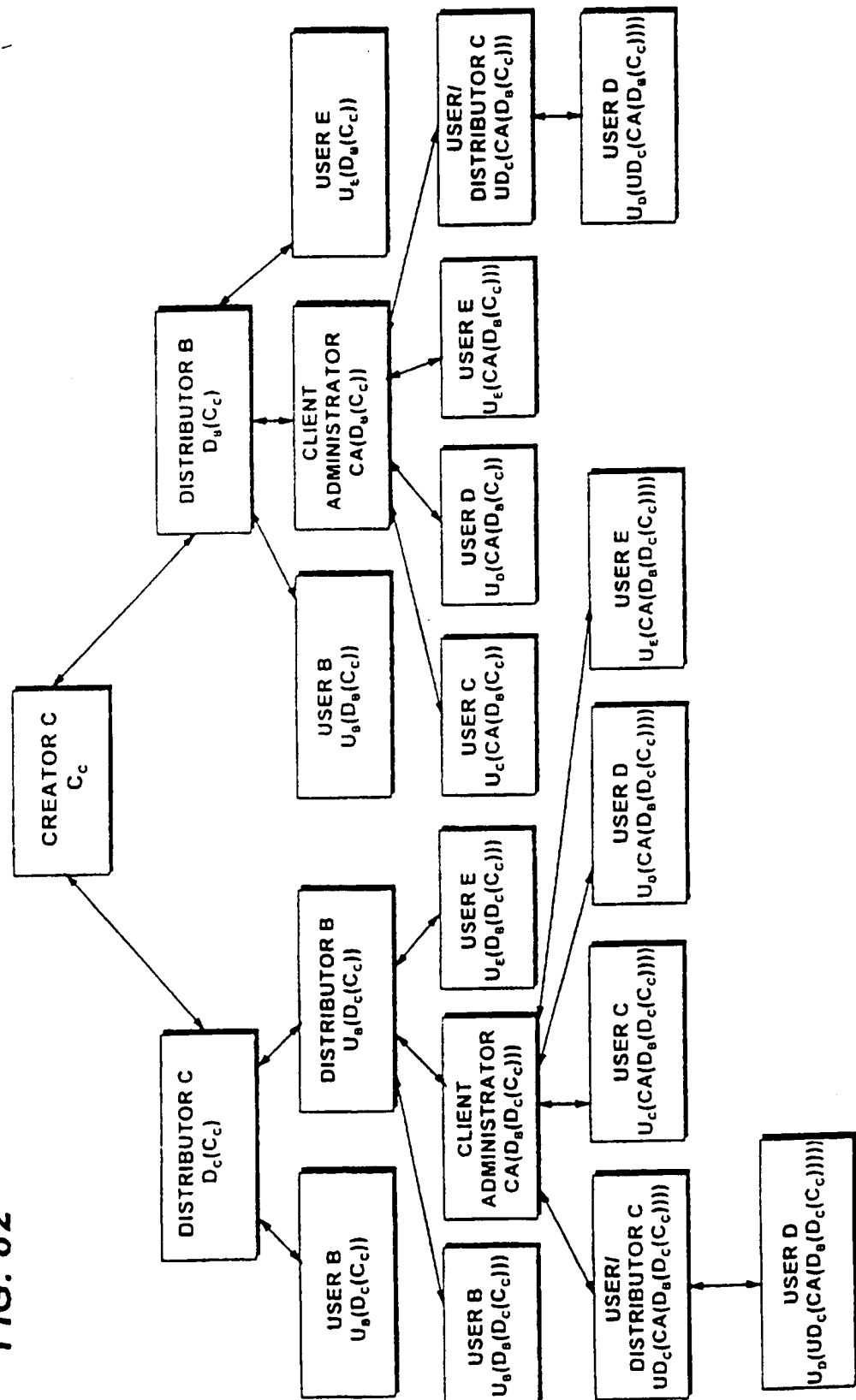


FIG. 81

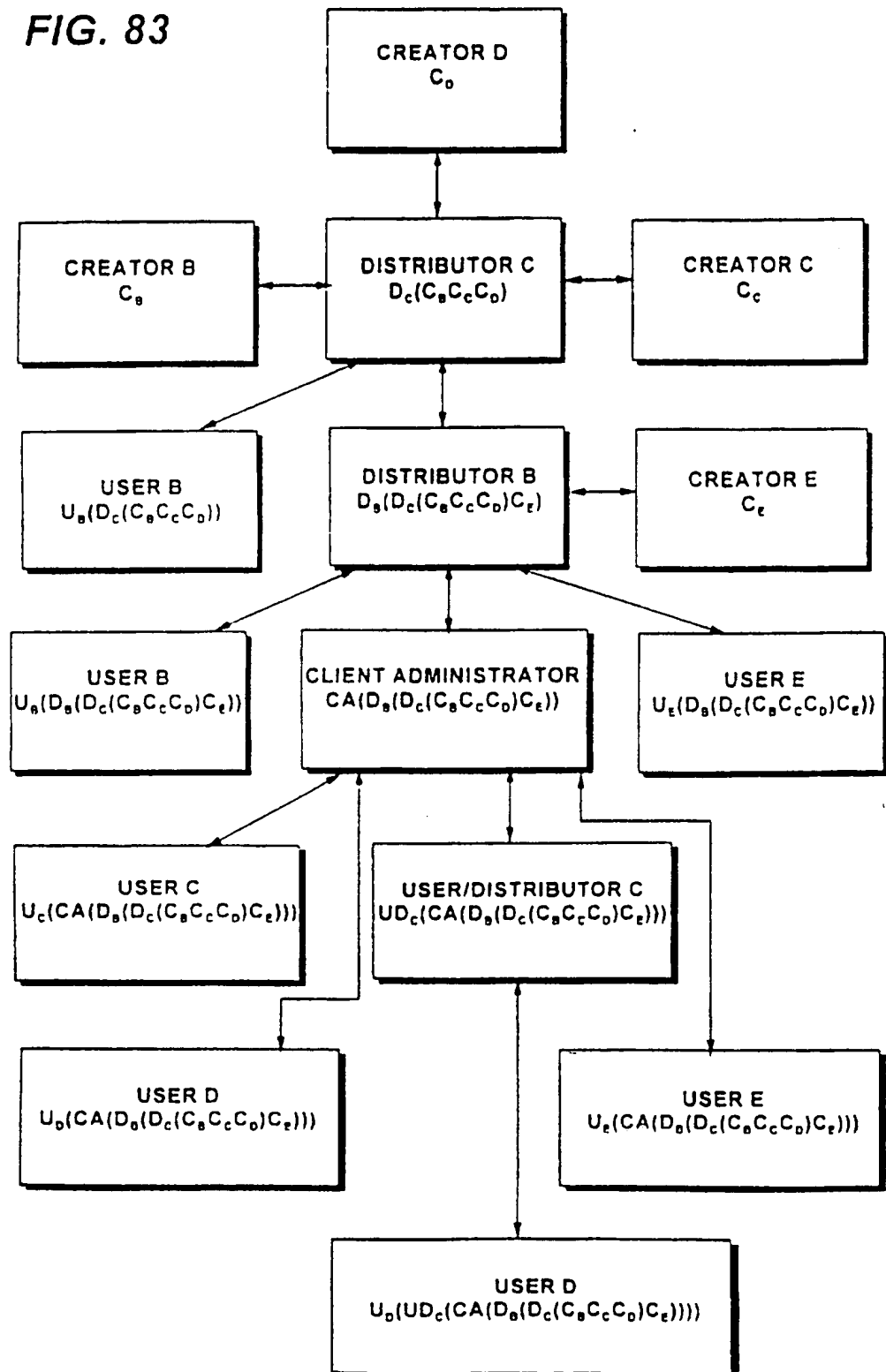
**FIG. 82**





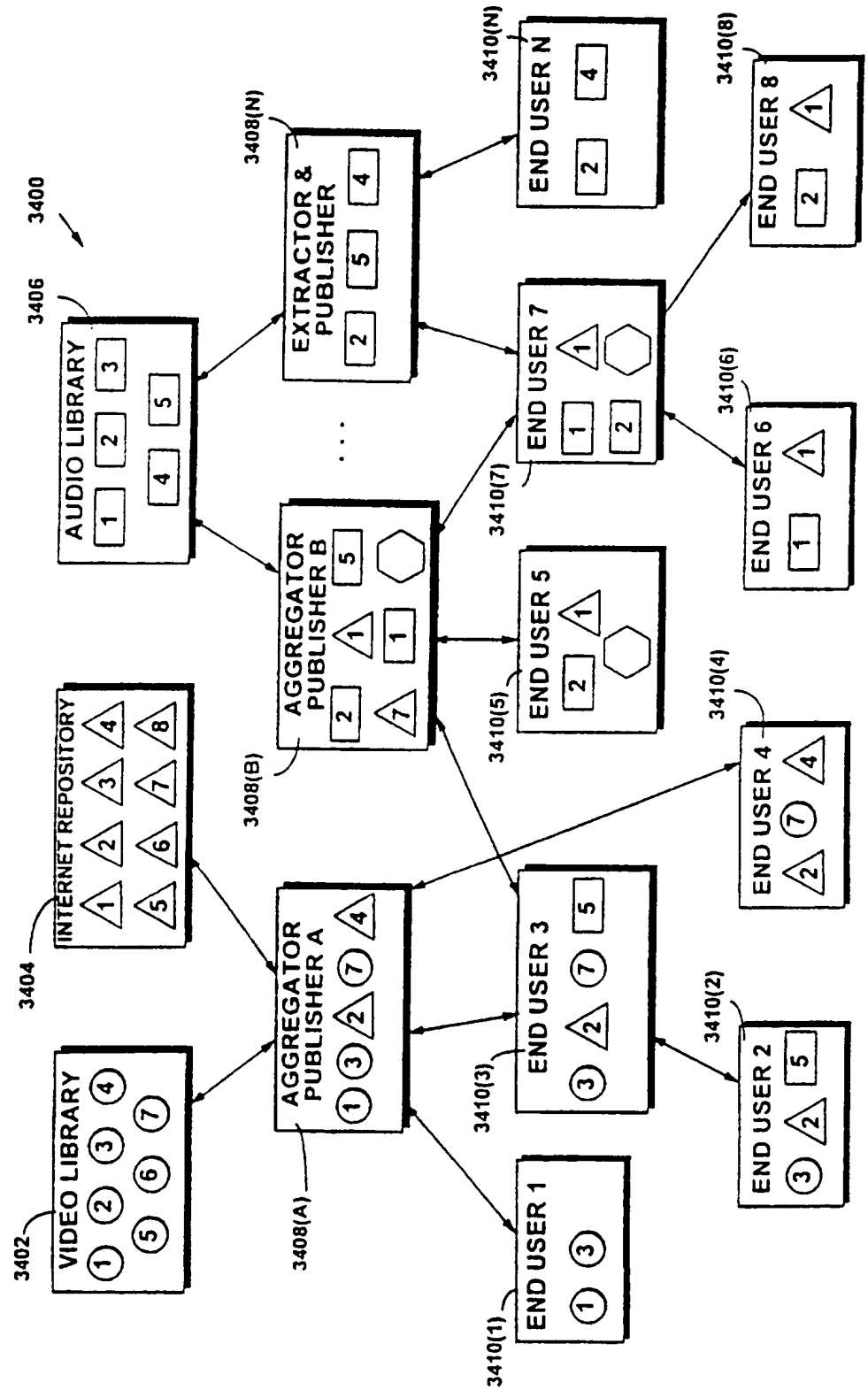
159/163

FIG. 83



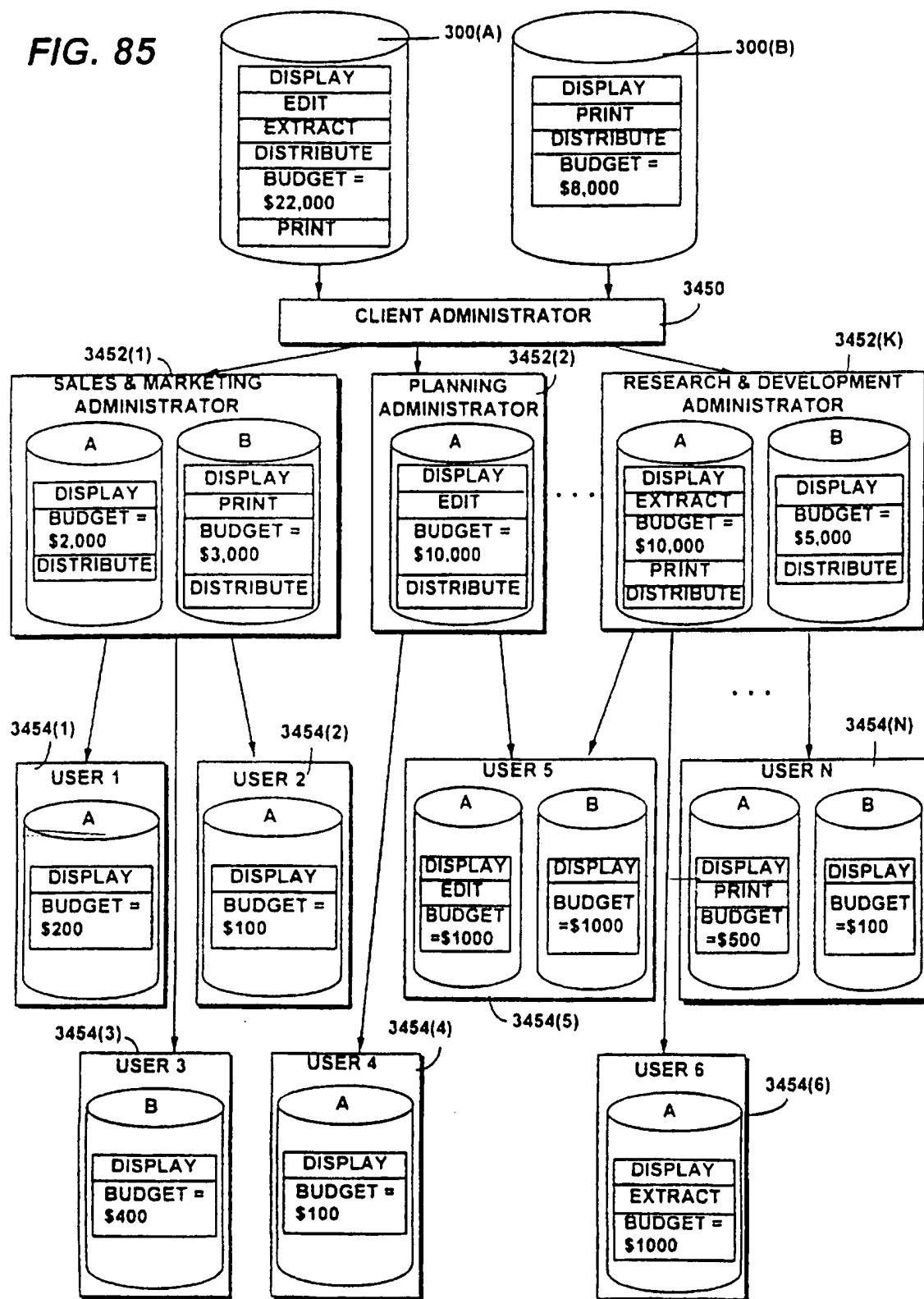
160/163

FIG. 84

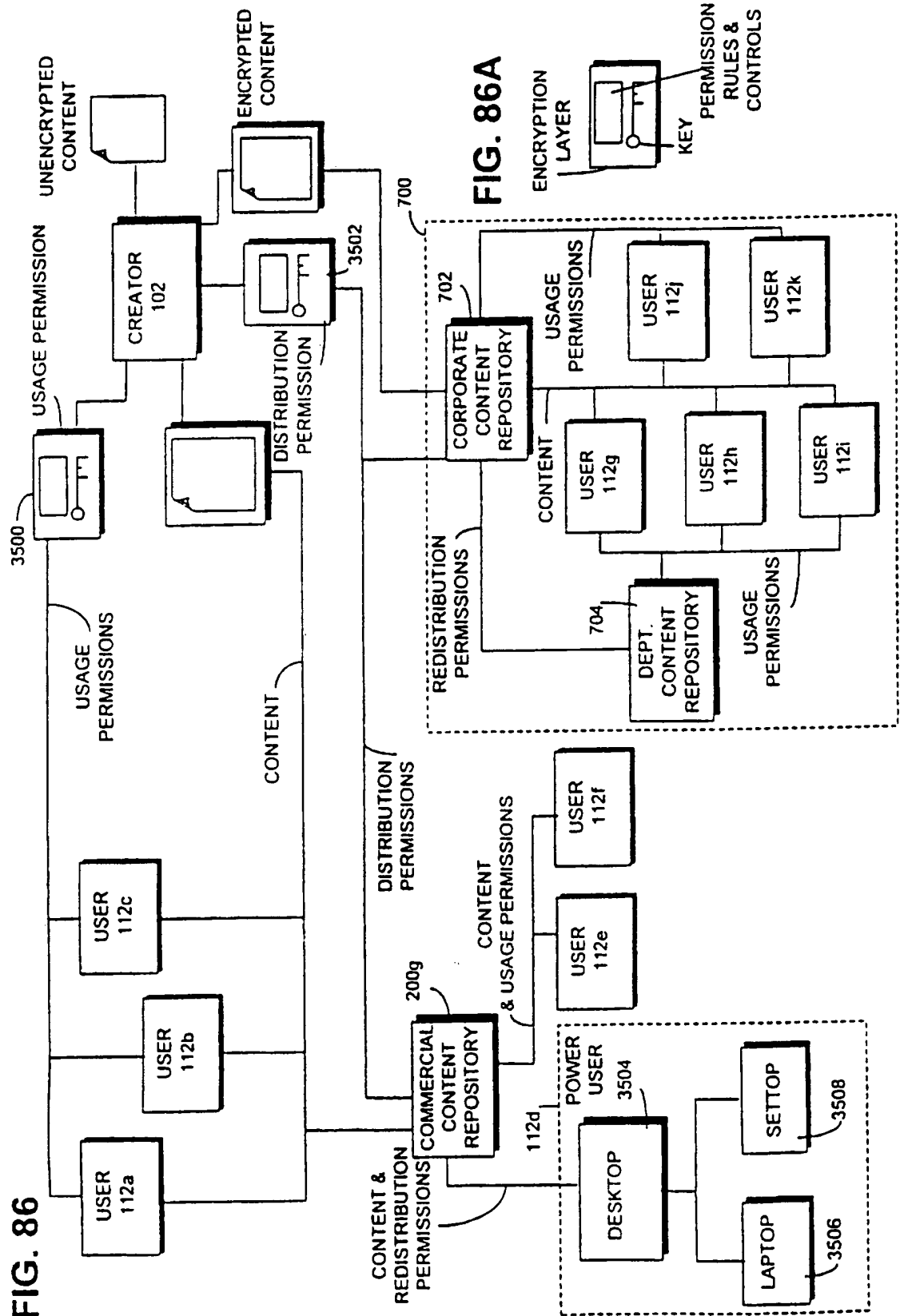


161/163

FIG. 85



162/163



163/163

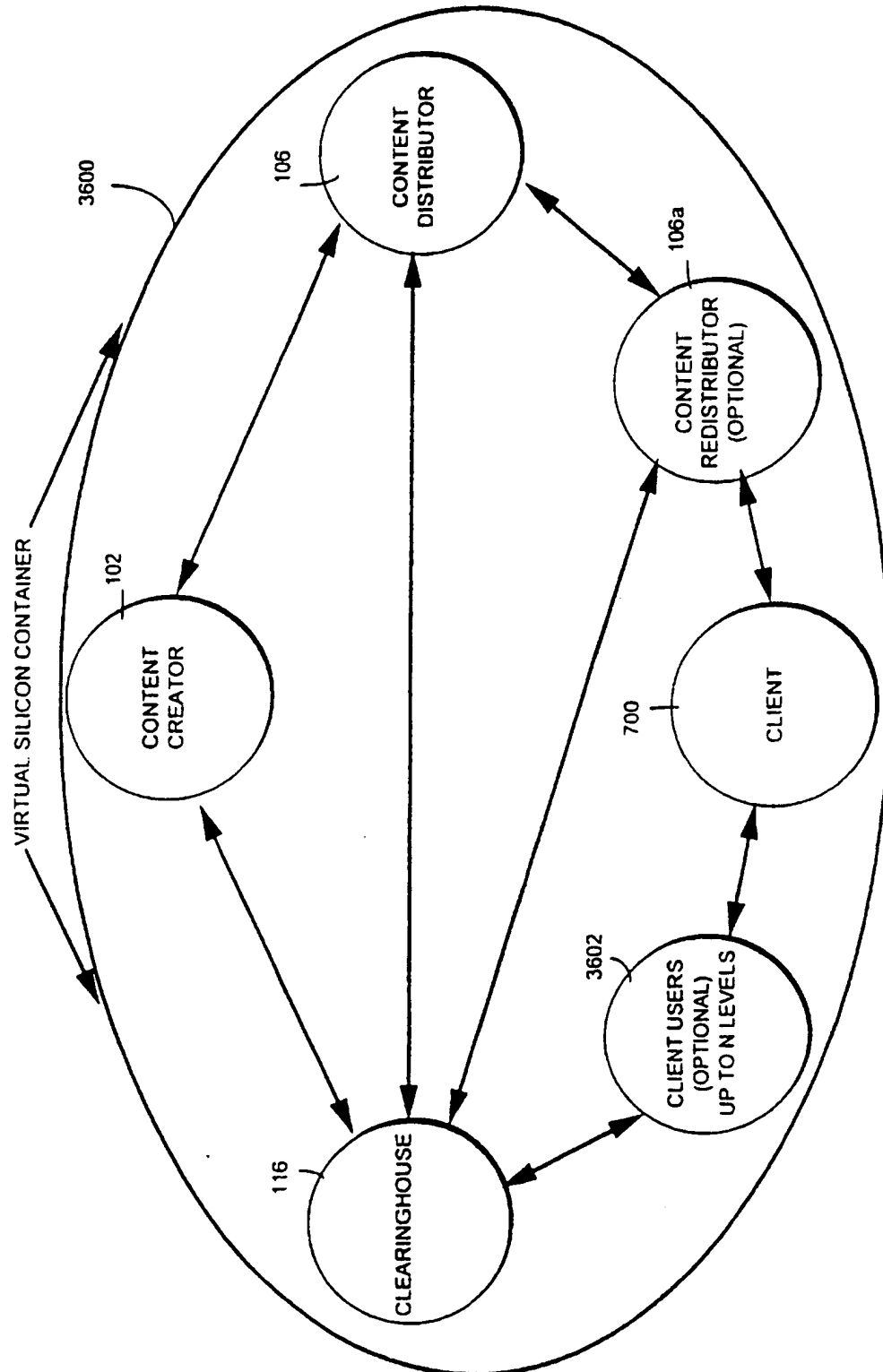


FIG. 87

# INTERNATIONAL SEARCH REPORT

PCT/US 97/15243

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 6 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CHOUDHURY A K ET AL: "COPYRIGHT PROTECTION FOR ELECTRONIC PUBLISHING OVER COMPUTER NETWORKS" IEEE NETWORK: THE MAGAZINE OF COMPUTER COMMUNICATIONS, vol. 9, no. 3 May 1995, pages 12-20, XP000505280 see the whole document ---	18
Y	WO 90 02382 A (INDATA CORP) 8 March 1990 see abstract; figures 2,12,13,15 see page 18, paragraph 3 - page 21, paragraph 2 see page 23, last paragraph - page 24, paragraph 1 ---	1-10
A	---	11-17
	---	---

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

### \* Special categories of cited documents:

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- \*S\* document member of the same patent family

Date of the actual completion of the international search

17 December 1997

Date of mailing of the international search report

29/12/1997

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Powell, D

## INTERNATIONAL SEARCH REPORT

PCT/US 97/15243

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
Y	EP 0 715 246 A (XEROX CORP) 5 June 1996 see the whole document	1-10
A	-----	11-17
A	US 5 224 163 A (GASSER MORRIE ET AL) 29 June 1993 see the whole document	11-16
A	-----	
A	WO 94 01821 A (SECURE COMPUTING CORP) 20 January 1994 see the whole document	17
A	-----	
A	WO 94 03859 A (INT STANDARD ELECTRIC CORP) 17 February 1994 see the whole document	17
	-----	

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9002382 A	08-03-90	AU 4188289 A	23-03-90
		EP 0472521 A	04-03-92
		US 5247575 A	21-09-93
EP 0715246 A	05-06-96	US 5638443 A	10-06-97
		JP 8263439 A	11-10-96
US 5224163 A	29-06-93	NONE	
WO 9401821 A	20-01-94	US 5596718 A	21-01-97
		AU 663406 B	05-10-95
		AU 4672693 A	31-01-94
		EP 0649546 A	26-04-95
		JP 7509086 T	05-10-95
WO 9403859 A	17-02-94	EP 0606401 A	20-07-94
		JP 7502847 T	23-03-95